003_CWRK: Final Project Report

Final Report

701030_A24_T12

14th May 2025

by

LOBONT, LEONARDO

LOUGHREY, NATHAN

PORTAS-JONES, HARVEY-JAY

RANCE, EDWARD

Abstract

This report details progress on an arm exoskeleton for stroke rehabilitation controlled by a 14-sensor EEG headset. The project has advanced with adjustments to the plan, focusing on hardware prototyping, safety and testing to ensure usability and reliability.

Table of Contents

List of Tables

Table of Abbreviations

Abbreviation	Definition
EEG	Electroencephalogram
PDD	Project Definition Document
CFRP	Carbon Fibre Reinforced Polymer
PLA	Polylactic Acid
BCI	Brain-computer Interface
USB	Universal Serial Bus
PWM	Pulse-Width Modulation
ROS	Robot Operating System
MQTT	Message Queuing Telemetry Transport
CPU	Central Processing Unit
GPIO	General-Purpose Input Output
FOS	Factor of Safety

Table of Formulas

Formula	Definition
$\sigma = \frac{My}{I}$	Bending Stress
$I = \frac{bh^3}{12}$	Moment of inertia in regular cross section
$\sigma = \frac{F}{A}$	Normal stress
$P = \frac{4M_{max}}{L}$	Breaking load from max moment
$E = \frac{\sigma}{\varepsilon}$	Young's modulus
$FoS = \frac{Strenght}{Load}$	Factor of Safety (FOS)

Table of Figures

Introduction

This project aims to develop a rehabilitative exoskeleton arm capable of wrist rotation and bicep curl, controlled via electroencephalogram (EEG) signals from the Emotiv EpochX 14-channel headset. While the original Project Design Document (PDD) proposed custom EEG signal processing using MATLAB and Simulink, this was reconsidered due to restrictions on accessing raw EEG data. As a result, signal interpretation is handled through Emotiv's proprietary Cloud platform. This adjustment is appropriate for the project's current scope as a proof of concept demonstrating the feasibility of EEG-based rehabilitation control systems.

Although the implementation has evolved, the project's core structure and objectives remain aligned with the original plan. Early in development, it became clear that running the complete control stack, including motor commands and signal parsing, on the Jetson Nano was not feasible due to hardware and resource limitations. Specifically, three of the four Jetson Nanos provided had non-functional I^2C buses, and one exhibited central processing unit (CPU) instability, preventing consistent performance. The need for more straightforward electrical integration and real-time control prompted a revised architecture: control logic in the final design is distributed across three Arduino boards, each responsible for motor control, sensor feedback, and interface logic, while the Jetson Nano functions as a central coordinator.

Mechanical design also shifted from carbon fibre reinforced polymer (CFRP) to polylactic acid (PLA), motivated by budget limitations and the desire to use more accessible materials for rapid prototyping. These refinements allowed faster iteration and testing without compromising the overall goals: creating a portable, safe, EEG-driven assistive device for post-stroke arm rehabilitation. The system is being evaluated based on the ease of user interaction with Emotiv's mental command training interface, which translates cognitive intent into discrete actuation commands.

Requirements and Evaluation

This section outlines how the system's requirements evolved and how each was addressed. It covers key changes to the original plan, revised objectives and validation through testing. Together, these demonstrate the feasibility and effectiveness of the final prototype.

Changes of Requirements

The initial requirements outlined in CWRK-001 included assumptions that the team would have full access to raw EEG data from the Emotiv headset and that all signal processing and control logic could be managed directly on the Jetson Nano. These assumptions were not fully validated in early planning and were not formally listed as project risks, representing a gap in the initial risk assessment process. Earlier versions of the report did not explicitly account for potential limitations of the Emotiv platform, nor the possibility that raw EEG access might be restricted. This limitation should have been identified and discussed more thoroughly in the early requirements stage.

Ultimately, access to raw EEG data was restricted due to the proprietary nature of the Emotiv platform. This removed the requirement to develop and validate a custom EEG signal processing algorithm. Instead, the project adopted Emotiv's cloud-based mental command processing, which offers pre-defined cognitive classifications. At the same time, this limited customisation was acceptable for the project's proof-of-concept aims.

Secondly, although initial planning treated the Jetson Nano as the central controller, computational and interfacing limitations became more evident during integration. While Emotiv's platform handled cognitive signal processing externally, the Jetson faced issues related to hardware reliability (e.g., three of the four units had non-functional I²C buses, and one had CPU instability), as well as practical difficulties managing general-purpose input output (GPIO) and real-time motor control. This and the desire to modularise the system for easier debugging and safety led to the decision to adopt a distributed architecture. The updated system delegates low-level motor control and sensor feedback to three Arduino boards, while the Jetson Nano serves as a high-level coordinator and interface with Emotiv.

Hardly significant delays in setting up and running simulations, mainly due to time constraints and software compatibility issues, meant that simulation-based verification and associated performance requirements were removed from the scope. This was a pragmatic decision to prioritise physical prototyping and hands-on testing in the available timeframe.

Lastly, the initial requirement for a lightweight composite frame was revised due to material availability and lack of access to CFRP. PLA was selected as a more accessible and affordable alternative. While this material was sufficient for initial testing, one prototype failure highlighted the importance of print settings—functional strength was achieved only when printed at an appropriate infill density, layer height and position (see "Mechanical Analysis of Parts" section).

Summary of Requirements

This project aims to develop a proof-of-concept EEG-controlled exoskeleton for rehabilitation, specifically targeting bicep curl and wrist rotation functionality. The system must safely translate user intent into physical motion using a 14-channel Emotiv Epoc X EEG headset. It must operate reliably, be safe for users, and allow for limited degrees of freedom, making it suitable for rehabilitation scenarios. The exoskeleton must be wearable, portable and comfortable for extended use.

The core functional and safety requirements are as follows:

1. **Accurate EEG Signal Interpretation** – The system must recognise basic motor-cortex imagery commands, such as left or push, via the Emotiv cloud services and then translate the data into mechanical action.
2. **Real-Time Actuation**—The system must actuate two servo motors in real time based on EEG input with minimal latency.
3. **Power and Control Reliability—**The robotic arm must operate independently and stably, using INA226 current sensors and dual 12v 6a batteries in parallel.
4. **Safe Operation—**To prevent unintended motion, the system includes an emergency stop button, limit switches, an INA226 current cutoff, built-in overcurrent protection in the servo motors, and software shutdown.

5. **Modular Architecture—**Hardware must be accessible and modifiable. The final setup includes 3 Arduino Unos for decentralised control and a Jetson Nano as the system's processing hub and Wi-Fi connection.
6. **Mechanical Feasibility** – The design must support limb movement without structural deformation. PLA parts are printed with adequate infill and tested under torque.
7. **User Feedback Integration—**Simulation software provides real-time system status, signal strength integrity, and training proficiency.

These requirements are derived from the need to validate EEEG signal control in neurorehabilitation devices. Emphasis is placed on modularity, safety, affordability and demonstrating viability using consumer-accessible components.

Justification of Changes

The original plan relied on the Jetson Nano for I2C communication and robot operating system (ROS) topic control. However, hardware issues, including I2C instability and a damaged Jetson unit, made this unfeasible. Control was redistributed to three Arduino Unos, each handling specific tasks via a universal serial bus (USB) serial. This improved reliability, reduced wiring complexity and enabled real-time processing.

The team initially intended to use ROS and Gazebo for system integration and simulation. However, as development progressed, it became clear that ROS introduced significant complexity and overhead disproportionate to the project's scope, especially as simulation of the physical environment was ultimately deprioritised in favour of real-world testing. This led to a switch to message queuing telemetry transport (MQTT) and Node-RED, which enabled a more modular and lightweight architecture using isolated Python scripts. This approach improved fault tolerance and simplified debugging during hardware integration.

Due to license changes in Emotiv's software and firmware [Emotiv, n.d.] Node-RED was moved from containerisation to a bare-metal deployment to maintain compatibility. However, the MQTT broker remained containerised and unaffected by the licensing constraints. Note that our original intention was to keep the software contained for ease of use and control since we were running the software on a university-provided laptop.

These changes addressed hardware limitations and increased system stability without affecting core functionality.

Demonstrating the Fulfilment of Requirements

Each realised requirement was tested and verified through targeted evaluations during the integration and prototyping phases:

1. **EEG Signal Interpretation**
   EEG mental command recognition was tested using the Emotiv Epoc X and its cloud-based software. The team verified that two distinct commands (e.g. left/right imagery) triggered appropriate motor responses via Node-RED and MQTT logging.
2. **Motor Actuation (Bicep and Wrist)**
   DServo 35kg servos were integrated into the PLA printed frame. Controlled tests confirmed consistent movement under command without stalling, using predefined EEG signals to trigger actuation.
3. **Power System and Sensor Monitoring**
   The INA226 current sensor was tested using controlled motor loads. The output was read through the corresponding Arduino and monitored via serial output on the Jetson Nano. Reading confirmed that the current stayed within safe thresholds.
4. **Safety Systems**
   Emergency stop functionality was verified by cutting relay power to the system mid-operation. Four limit switches were manually triggered to confirm motion cut-off at physical boundaries.
5. **Modular Software Architecture**
   Each subsystem was run independently through separate Arduino boards and managed via modular Python scripts. The use of MQTT topics allowed isolated testing and fault handling.
6. **System Portability and Usability**
   The 3D printed design, supported by a shoulder strap and powered by two 12v 6A batteries, was tested for comfort and continuous operation. No structural failure or overheating was observed during testing.

All fulfilled requirements were validated through real-time testing rather than simulation, which aligned with the project's revised scope. The only section lacking was the wrist, which was supposed to be a 90-degree turn; however, we only achieved a 45-degree total left and right. This overdoes affect the proof of concept being achieved; if the team had access to more time and resources, this could have easily been achieved and will be discussed below.

Explanation of the met Requirements

The following is a summary of how each key requirement was met and validated during development and testing:

- **EEG Command Recognition**
  Mental commands were captured using the Emotiv Epoc X headset. The cloud platform processed the signals and sent structured outputs to the Jetson Nano. Functionality was confirmed through consistent and repeatable control of the servos in response to mental input.

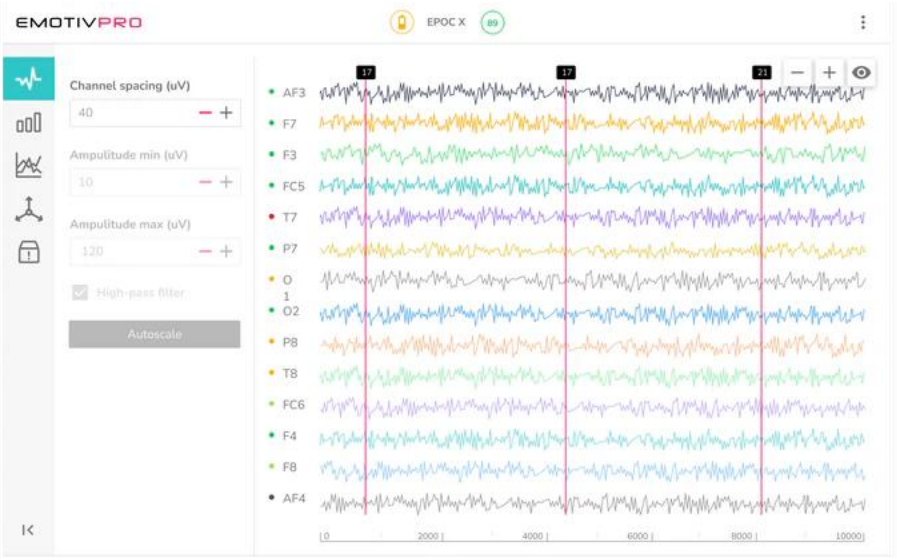


Figure 1 - EEG Command Recognition

- **Bicep Curl and Wrist Rotation**
  Motor control was successfully implemented using two DServo 35kg servos. EEG-triggered actions (left and right wrist rotation and up and down bicep curl) moved the exoskeleton arm through its two degrees of freedom (SV-T04, D-T03). Movement was repeatable within the physical design limits (WS-T05, SV-T04, SV-T07, D-T04, D-T08).

Figure 2 - Bicep Curl and Wrist Rotation

- **Safe Operation (Emergency Stop and Limit Switches)**
  The emergency stop function was tested successfully by cutting power via the relay via software (WS-T06, SF-T04, I-T01, I-0T03) . A physical emergency stops the power to the servos, too (I-T07) . Limit switches were triggered manually and during motion to halt movement as designed. However, the software range of the motors does not reach the limit switches (SF-T03); this is just an additional feature to ensure the user's safety if the servos malfunction. All safety responses were reliable and immediate (ELC-T03).

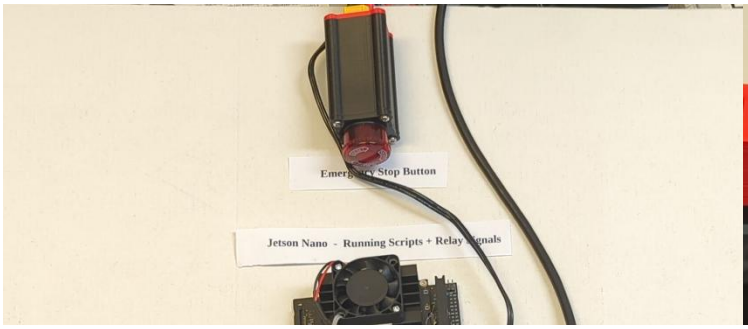


Figure 3 - Hardware E-Stop

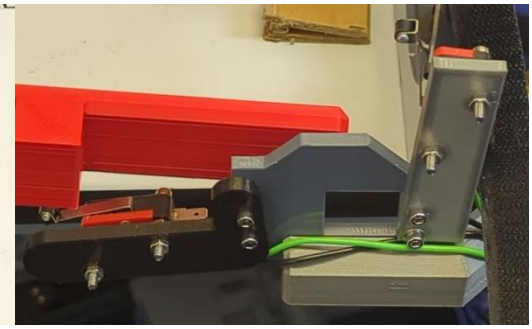


Figure 4 - Limit Switches

- **Electrical Monitoring (INA266)**
  The current draw was monitored using the INA226 chips connected to an Arduino. Sensor values were received by the Jetson Nano via serial and monitored using MQTT logs (WS-T04, WS-T05, WS-T06, WS-T07) to ensure the current remained within safe operating levels (SF-T01, ELC-T06).
- **System Architecture and Robustness**
  The split between Jetson Nano and Arduino proved effective. Each Arduino handles its own I/O tasks independently. The modular Python-MQTT system allowed fault isolation and debugging without complete system disruption (JN-T05).
- **Portability and Structure**
  The structure was printed in PLA and tested under load with servo actuation (M-T01). The system will be worn using a shoulder strap and function as a self-contained, battery-powered prototype, meeting user comfort and mobility design goals (M-T04). A PDU was created to begin with; however, due to time constraints and a multitude of issues, the team decided to abandon this idea. However, it was achieved for the first prototype.

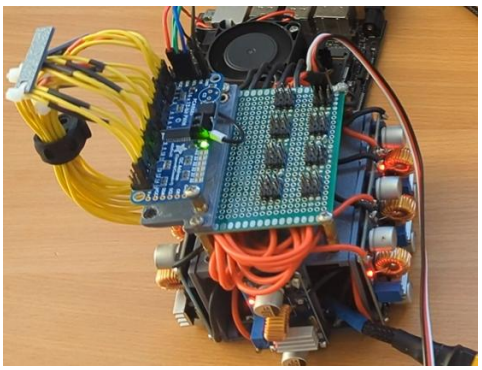


Figure 5 - Power Distribution Unit

These requirements were demonstrated through live testing, debug logs or hardware validation and contributed directly to the system's overall proof-of-concept success.

Unmet Requirements

The only significant requirement not fulfilled was the full simulation of the exoskeleton using Gazebo. This was initially intended to support control validation and integration with ROS. However, progress was limited due to time constraints and software compatibility issues.

This shortfall did not impact the final system's physical implementation or core functionality, as all control and testing were successfully handled using the actual hardware and an MQTT-based communication flow. The decision to deprioritise simulation was necessary to focus resources on real-world testing and system reliability, which will be discussed below.

Tests

This section outlines the tests to verify system functionality, safety and reliability. Each test group targets a specific subsystem, ensuring the prototype meets its intended use in real-world rehabilitation scenarios

Table of Tests

Test ID	Test name	Test description	Test pass criteria	Pass or Fail
EEG Tests (E-T Series)				
These tests verify the functionality and reliability of the Emotiv EEG headset. They cover essential actions such as charging, powering on, connecting via Bluetooth, interacting with software (Emotiv Suite), saving trained profiles, and ensuring the EEG signal's consistency. Importantly, they assess whether user training sessions are practical and not overly fatiguing, which supports long-term usability in rehabilitation contexts.				
EEG				
E-T01	Headset charges	Headset charges and stores power	Charges, stores power	Pass
E-T02	Power on the headset	The headset powers on	Turns on	Pass
E-T03	Bluetooth connection	A Bluetooth connection is established between the Windows system and the headset	Connects to the pc successfully	Pass
E-T04	Connect to the software	Emotiv Suite connects to the headset	Connects to the software successfully	Pass
E-T05	Sends signals to Windows system	Signals received in the Emotiv suite	Signals are received	Pass
E-T06	Can interact with the software training suite	Signals can be used to control in-software training suites	Signals cause responses in software, e.g. cube	Pass
E-T07	Trained model saved as profile	Can reopen and access a trained dataset	Can access the created profile for the trained data	Pass
E-T08	Headset signals are consistent	Checking the signals to make sure that they align with the desired results	Signals show similar patterns when put through a controlled environment	Pass
E-T09	User Fatigue	Understanding the best sets for training so a user does not get too tired and too frustrated	Two 15-minute sets within an hour	Pass

Windows System Tests (WS-T Series)

This group of tests confirms that the software environment on the Windows machine is correctly configured. It ensures the containerised services (Docker), MQTT broker (Mosquitto), and flow-based development tool (Node-RED) are operational. It also confirms that signal flow between the EEG headset and the MQTT ecosystem is functional and secure, even across system firewalls.

WS-T01	Docker installed	Docker is installed and running on WSL	Docker is installed and can be opened	Pass
WS-T02	Mosquitto is installed and running	Mosquitto is running and configured as a server in Docker	Mosquitto server running to connect to	Pass
WS-T03	Node-Red is installed and running	Node-Red running in Docker	Node-RED is accessible in the browser	Pass
WS-T04	Node-Red connects to the MQTT broker	Node-Red connected to the MQTT broker for communications	Node-Red can connect to an MQTT server	Pass
WS-T05	Signal sent and received through the firewall	Signals from Emotiv can be sent and received through the firewall	Node-Red can receive the sent commands	Pass
WS-T06	Signal propagated via MQTT	Signals can be propagated through MQTT via Node-Red	Node-Red can send messages of commands via MQTT	Pass
WS-T07	Status messages received by MQTT	Status messages can be received into Node-Red	Status feedback can be received	Pass

Jetson Nano/Raspberry Pi Tests (JN-T Series)

These tests were originally intended to validate the Raspberry Pi 5 as a control unit. They check power-on behaviour, script execution in C++, and communication with the MQTT broker and motor controller. Though the Pi was not ultimately used in the final system, these tests document an important phase of development and fallback exploration before returning to the Jetson Nano.

JN-T01	Jetson Nano powers via barrel jack.	Jetson powers on and boots	Jetson powers on and boots	Pass
JN-T02	Python libraries and compiler installed	Python and dependencies are installed	Python is installed and can run with the necessary dependencies	Pass
JN-T03	Scripts running	A Python Programme can run	Python scripts run successfully	Pass
JN-T04	Connected to the MQTT broker	The Python script connects to the MQTT broker for commands	The Python MQTT broker can connect successfully	Pass
JN-T05	Signals received	MQTT signals received	Signals are received from Node-Red via MQTT	Pass
JN-T06	Signals sent to the motor controller	Signals sent to the motor controller	The motor controller receives the signals for controlling the servos	Pass
JN-T07	Status messages sent by MQTT	Status messages sent back via MQTT	Status messages can be sent back to Node-Red for status updates/checks.	Pass
JN-T08	PCA9658 test	Does the PCA9658 communicate with the Arduino and Jetson	Jetson receives communications from the PCA9658	Pass

Servo Tests (SV-T Series)

These tests assess the behaviour of the servo motors that move the exoskeleton. They ensure accurate motion control, enforce software and physical movement limits, and check for stable holding strength and safe load bearing. These are critical for confirming that the system performs safe and smooth limb movements.

SV-T01	Moves when signals are received	When the signal from the motor driver is received, the servos move to the correct position	Motor moves to the position it is told to using PWM	Pass
SV-T02	Stops at software limits	Software limits prevent over-rotation in isolation	Servo can't pass limits	Pass
SV-T03	Stops at hardware limits	Hardware limits prevent over-rotation in isolation	Servo can't pass limits	Pass
SV-T04	Has smooth motion in small steps	Small incremental steps when moving to allow smooth motion	Movements are small but consistent	Pass
SV-T05	Doesn't move when not told to	When no input is sent, the servo stays stable and doesn't move	The servo only moves when told to	Pass
SV-T06	Has the holding strength to catch the arm, and either keep it or slowly lower it	Can hold an appropriate load and let an excessive load down slowly if necessary	The load-bearing capacity is adequate to support the average arm	Pass
SV-T07	Homes to "origin" at the start and end of the operation	Homes are ready for putting on and storage	Servo homes when the system turned on and off	Pass
SV-T08	Handles the Weight of a factor of 1.8			Pass

Safety Feature Tests (SF-T Series)

Focused on protecting users and hardware, these tests verify that emergency stop mechanisms work mechanically and via software. Over-current protection and movement limit enforcement are also validated to reduce the risk of user injury or hardware failure.

SF-T01	Over-current protection cuts power	Over-current protection cuts power to the relevant components	Power gets cut where necessary for safety measures to prevent operation	Pass
SF-T02	Mechanical limits prevent over-rotations	Mechanical limits prevent over-rotation in a partially completed system	Servos cannot move outside their limits	Pass
SF-T03	Software limits prevent over-rotations	Software limits prevent over-rotation in a partially completed system	Servos cannot move outside their limits	Pass
SF-T04	Software emergency stop works in isolation	Software emergency stop prevents the passing of commands	Emergency stop halts command propagation	Pass
SF-T05	Mechanical emergency stop works in isolation	Mechanical emergency stop prevents operations	Stops system operation within the necessary modules	Pass

Integration Tests (I-T Series)

These tests validate full-system functionality across all modules. They confirm that messages from the EEG headset traverse the signal pipeline correctly and that all components (headset, server, actuators, UI) remain in sync. Crucially, safety features are tested within the full operational context, not just in isolation.

I-T01	Messages are passed reliably around the system	Messages are passed through the pipeline with maintained integrity	The message that is sent first is received last, with all necessary information intact and correct	Pass
I-T03	Commands from the headset are followed	Commands sent are followed and interpreted into movement appropriately	The movements match the expected operations	Pass
I-T04	Commands being sent do not override safety limits	All safety limits are followed in an integrated system using all systems	Safety limits are followed, cuts off and stops where required	Pass
I-T05	Status commands reach the Windows system to enforce limits via software	Status messages are propagated back to the Windows system to display status and enforce limits	Status is sent back through the system and is used to show and operate with clear intent	Pass
I-T06	Emergency stops (Software) prevent operation, and the status is sent back	When emergency stops are active, the status shows as such while the operation is ceased	The emergency stop notification shows	Pass
I-T07	Emergency stops (Hardware) prevent operation, and the status is sent back	When emergency stops are active, the status shows as such while the operation is ceased	The emergency stop notification shows	Pass
I-T08	Disconnections pause operation safely.	When a disconnection occurs, the exoskeleton stops operating safely for removal or awaits reconnection	Operation ceases to prevent negligent movements and prevent damage and harm to users and property.	Pass
I-T09	Power Supply and Distribution unit testing	Test the PDU and PSU to ensure they properly distribute power across all components, including the Jetson Nano, Arduino, and servo motors.	Ensure no overheating or power instability occurs when the system is under load.	Pass

Electronics Tests (ELC-T Series)

These tests ensure electrical system integrity, including correct wiring, functional power distribution, thermal safety, and the system's ability to tolerate stress. They also verify mitigation strategies for faults like solder bridges or connector failures, making them essential for long-term hardware reliability.

ELC-T01	Incorrect Wiring	Incorrect wiring, for example, a relay being wired into the I2C bus	Works as intended, a mitigation would be to draw/write up schematics and circuit diagrams	Pass
ELC-T02	PDU Testing	PDU “Doesn’t Work” as in a multitude of physical component issues we can’t direct	Pre-test every component within the PDU to allow us to narrow down the issue	Pass
ELC-T03	Debugging Issues	Following ELC-T03, we can use Python scripts for testing and Python for our final program.	It helps solve potential issues. If we know there are no hardware issues, it's a software issue, and vice versa.	Pass
ELC-T04	C Connector Damaged	Is our C connector to our PI-5 damaged? Use GPIO as a power route to test	Can the PI be powered in other ways? Yes? Pass the test.	Pass
ELC-T05	Soldering Bridges	Ensuring careful planning, visual inspection, testing with a continuity multimeter, and cleaning the board frequently during building	If there are no short circuits and the description has been followed, and the system turns on, pass the test.	Pass
ELC-T06	Correct Flow	Measure the voltage and current output of the overall PDU after the ELC-T02 and the build process.		Pass
ELC-T07	Over-Heating	Thermal testing, proper spacing of components and heat sink on pi-5		Pass
ELC-T08	Full System Test	Overload the system and “stress” it out to test its overall capabilities under the maximum loads a user of the arm could output		Pass

Materials				Pass

Materials Tests (M-T Series)

These tests evaluate the exoskeleton's mechanical design. They confirm that 3d-printed parts are strong enough, that pivot joints and straps function correctly, and that the assembled structure fits the user comfortably and securely.

M-T01	3d prints are strong enough to take the weight and forces applied to them	Test if the parts can withstand loads over the expected	Parts do not break under anticipated and within reasonable overloaded bounds	Pass
M-T02	Materials have low enough friction to move around or on each other reliably.	Test if sliprings and pivots can move smoothly when actuated by servos	Parts move smoothly and consistently	Pass
M-T03	Straps have enough rigidity to hold the exoskeleton in place	Test if, when under operation, the straps are rigid enough not to let the exoskeleton hang loosely or move unexpectedly	Straps hold the exoskeleton firmly in place to prevent movements in unintended directions	Pass
M-T04	The design fits on the arm	Test if the parts fit onto someone's arm once assembled	Someone can fit into the design, and it is comfortable.	Pass

Dynamics Tests (D-T Series)

These tests simulate real-world operation by commanding arm and wrist motions under different conditions. They measure performance under repeated use, stress, weight load, and rapid switching between commands. They also assess how the system reacts to emergency stop inputs and how the user physically tolerates extended use.

D-T01	Raise (lifting) the arm	Can we raise the arm by 45 degrees to each point and hold the arm there	An arm reaches and maintains the target position without stuttering. No excessive strain on motors. Smooth descent.	Pass
D-T02	Wrist rotation	Can we rotate the arm to its maximum points and hold the wrist in place	The wrist rotates smoothly and accurately. The system does not overshoot the angle or struggle to hold position.	Pass
D-T03	Sequential movement test	Can we lift the arm 45 degrees, rotate the wrist, lift again and then return to the resting position	No conflicting commands or erratic movements. Movements transition smoothly without jerking.	Pass
D-T04	Resistance load test	Can the arm lift more than our subject's arm weight? (try on other people)	The arm holds steady without slipping. The system detects overload and prevents damage.	pass
D-T05	Rapid Switching	Can the project reliably switch between each movement? Speed is not required, but it will stress test the system.	No command is ignored or misinterpreted—no unexpected stalling or instability/locking.	Pass
D-T06	Emergency stop	Can the arm safely stop the arm whilst it is in motion	The arm halts immediately without dropping. Emergency stop does not lock the system permanently	Pass
D-T07	Project Fatigue	See if the project itself is reliable over extended use	The system remains responsive throughout. Minimal signal drift.	Pass
D-T08	User Fatigue	What is the average amount of time a user can use the arm for (maybe see how long it takes to "get used to it" too)	The user can use over the recommended time, which is 15 minutes	Pass

Explanation of Test Sections

EEG Tests (E-T Series)

These tests verify the functionality and reliability of the Emotiv EEG headset. They cover essential actions such as charging, powering on, connecting via Bluetooth, interacting with software (Emotiv Suite), saving trained profiles, and ensuring the EEG signal's consistency. Importantly, they assess whether user training sessions are practical and not overly fatiguing, which supports long-term usability in rehabilitation contexts.

Windows System Tests (WS-T Series)

This group of tests confirms that the software environment on the Windows machine is correctly configured. It ensures the containerised services (Docker), MQTT broker (Mosquitto), and flow-based development tool (Node-RED) are operational. It also confirms that signal flow between the EEG headset and the MQTT ecosystem is functional and secure, even across system firewalls.

Jetson Nano/Raspberry Pi Tests (JN-T Series)

These tests were originally intended to validate the Raspberry Pi 5 as a control unit. They check power-on behaviour, script execution in C++, and communication with the MQTT broker and motor controller. Though the Pi was not ultimately used in the final system, these tests document an important phase of development and fallback exploration before returning to the Jetson Nano.

Servo Tests (SV-T Series)

These tests assess the behaviour of the servo motors that move the exoskeleton. They ensure accurate motion control, enforce software and physical movement limits, and check for stable holding strength and safe load bearing. These are critical for confirming that the system performs safe and smooth limb movements.

Safety Feature Tests (SF-T Series)

Focused on protecting users and hardware, these tests verify that emergency stop mechanisms work mechanically and via software. Over-current protection and movement limit enforcement are also validated to reduce the risk of user injury or hardware failure.

Integration Tests (I-T Series)

These tests validate full-system functionality across all modules. They confirm that messages from the EEG headset traverse the signal pipeline correctly and that all components (headset, server, actuators, UI) remain in sync. Crucially, safety features are tested within the full operational context, not just in isolation.

Electronics Tests (ELC-T Series)

These tests ensure electrical system integrity, including correct wiring, functional power distribution, thermal safety, and the system's ability to tolerate stress. They also verify mitigation strategies for faults like solder bridges or connector failures, making them essential for long-term hardware reliability.

Materials Tests (M-T Series)

These tests evaluate the exoskeleton's mechanical design. They confirm that 3d-printed parts are strong enough, that pivot joints and straps function correctly, and that the assembled structure fits the user comfortably and securely.

Dynamics Tests (D-T Series)

These tests simulate real-world operation by commanding arm and wrist motions under different conditions. They measure performance under repeated use, stress, weight load, and rapid switching between commands. They also assess how the system reacts to emergency stop inputs and how the user physically tolerates extended use.

Updated Risk Assessment

Table of Risks

Risk ID	Risk Name	Severity * Likelihood	Score	Mitigation	Mitigated Severity * Likelihood	Residual Risk
	-Risk to Team Members					
RTM-001	Stress and Burnout	5*4	20	Follow the Gantt chart, taking weekends and holidays off	1*2	2
RTM-002	Illness	2*4	8	Have allocated sick days, weekend breaks and holidays in the plan/Gantt chart	1*3	3
RTM-003	Medical Constraints	5*2	10	Have allocated sick days, weekend breaks and holidays in the plan/Gantt chart	4*1	4
RIM-004	In proper us of GIT	5*3	15	Familiarise ourselves with GIT with practice projects and ensure branching	5*1	5
	-Risk to Derail the Project					
RDP-001	EEG Headset Not Working	4*4	16	Spare Headset or Switch Projects / Make sure it works VERY early	2*1	2
RDP-002	Poor Time Management	4*2	8	Keep referring to the plan and the Gantt chart alongside regular meetings	1*2	2
RDP-003	Diversion Externally	3*3	9	Refer to the plan and Gantt chart. Team can ground one another	2*2	4
RDP-004	Financial Constraints	3*3	9	We'll have to share the expenses	2*2	4

RDP-005	Compliance and Standards	5*2	10	Ensure that Safety and Ethics are upheld throughout the whole project	2*1	2
RDP-006	Scope Creep	5*3	15	Refer to the plan alongside weekly meetings and grounding one another	2*2	4
RDP-007	The Testing Environment is not suitable	5*2	10	Refine Algorithms for filtering noise	2*2	4
RDP-008	Disagreements	5*3	15	Discussions on changes and decisions during meetings to voice opinions professionally	2*2	4
RDP-009	Lack of Communication	5*2	10	Weekly meetings, outside communication and social apps	2*1	2
RDP-010	User Participation	2*5	10	The Team can test on themselves	2*6	6
RDP-011	Financial Constraints from the University	3*5	15	The Team can use their funds	2*2	4
RDP-012	Miscommunication between the Team and the Supervisor	3*3	9	Weekly meetings with the supervisor	2*1	2
RDP-013	Hard-disk Failure	5*1	5	GitHub Repository and Local Backups	1*1	1
RDP-014	Hardware Failure	4*3	12	The Team buys new components, or we use spares we can find	2*2	4
	-Risk to Users of the Product					
RUP-001	Electric Shock	5*3	15	Safety Testing with the Team	2*1	2
RUP-002	Injury to the Arm	5*2	10	Safety Testing with the Team	2*1	2
	-Electronic Risks					
RER-001	Pi-5 PolyFuse tripping/popping	5*4	20	Schematics, circuit diagrams, visual inspection, and voltage and current flow testing	5*2	10
RER-002	Incorrect Wiring	4*4	16	Schematics and circuit diagrams with a continuity multimeter testing	4*1	4

RER-003	Bridging Connections	5*4	20	Consistent cleaning during the build process and following the preplanned circuit diagram	5*1	5
RER-004	Over-Heating	5*3	15	PDU Case CAD design, using the SolidWorks to visualise the whole complete structure	4*1	4

What has changed?

Here's a summary of the most significant changes in the risk management plan, focusing on the most impactful risks:

1. Internal Risks (Team Member-related)

- **RTM-001 - Stress and Burnout:**
  Change: Emphasis has been placed on following the Gantt chart and taking regular breaks (weekends/holidays) to manage stress and burnout.
  What's Changed: Mitigation strategy refined, reducing residual risk from 20 to 2.
- **RIM-004 - Inappropriate Use of GIT:**
  Change: The team will familiarise themselves with GIT through practice projects and ensure proper branching practices.
  What's Changed: The risk mitigation was added, lowering the residual risk from 15 to 5.

2. Project Derailment Risks

- **RDP-001 - EEG Headset Not Working:**
  Change: The team now ensures the headset is tested very early, with a backup plan (spare headset or project switch) in place.
  What's Changed: Mitigation refined; residual risk dropped from 16 to 2.
- **RDP-002 - Poor Time Management:**
  Change: Focused on regularly referring to the Gantt chart and having consistent meetings to manage time effectively.
  What's Changed: The enhanced plan reduced residual risk from 8 to 2.
- **RDP-006 - Scope Creep:**
  Change: Weekly meetings and strict adherence to the plan to prevent the scope from expanding beyond what was agreed upon.
  What's Changed: Residual risk reduced from 15 to 4.
- **RDP-007 - Testing Environment Not Suitable:**
  Change: Refining algorithms for noise filtering as a mitigation to ensure the testing environment is viable.
  What's Changed: Residual risk lowered from 10 to 4.

3. Risks to Users

- **RUP-001 - Electric Shock:**
  Change: The team will conduct thorough safety testing within the group to ensure no electric shock hazards exist before user testing.
  What's Changed: Residual risk has been significantly reduced from 15 to 2.
- **RUP-002 - Injury to Arm:**
  Change: Similar to electric shock, safety testing within the team ensures that injury risks are managed early.
  What's Changed: Residual risk reduced from 10 to 2.

4. Electronic Risks

- **RER-001 - Pi-5 PolyFuse Tripping/Popping:**
  Change: The risk of fuse tripping has been mitigated by using detailed schematics and circuit diagrams and performing voltage/current flow testing.
  What's Changed: Residual risk reduced from 20 to 10.
- **RER-002 - Incorrect Wiring:**
  Change: Schematics, circuit diagrams, and continuity testing are emphasized in the mitigation.
  What's Changed: Residual risk reduced from 16 to 4.

Summary of Key Updates

- **Refined mitigations** focus on early detection, continuous monitoring, and specific actions (such as testing and adhering to plans).
- **Residual risk reduction** in critical areas like time management, EEG headset functionality, scope creep, and safety testing demonstrates improved preparedness.
- **New risk mitigation strategies** for GIT usage and project derailment have been implemented to improve project flow and prevent setbacks.

These changes ensure the project is more resilient and well-prepared for unforeseen challenges, with a focus on key risks that could significantly impact the project's success.

Design

This section presents the key design decisions behind the hardware, software and system architecture. It outlines how each element was developed to meet usability, modularity and performance project goals.

Bill of Materials

Arduino Uno R3	Emotiv Epoc X	Mosquitto (MQTT)	SolidWorks
Assorted M3 Fasteners	Foam Tape	Node-RED	Velcro Straps
Craft Foam	Hot Glue	PLA Filament	Windows Laptop
Docker	INA226 with 0.01Ω shunt resistor	PowerShell	Wire, solder and connectors
DServo 35kg	Jetson Nano	Python	
Emotiv Cortex	Limit Switch V-156-1C25	Relay SRD-05VDC-8L-C	

Emotiv BCI

The Emotiv Epoc X 14-channel EEG headset was used as the primary brain-computer interface (BCI) for capturing user-intended neural signals to control the robotic arm. The headset enables non-invasive monitoring of brainwave activity and transmits processed cognitive data to the system using Emotiv's proprietary software suite. The primary tools that we were able to utilise included:

- **Emotiv Pro** – for device connection, signal quality assurance, and real-time monitoring.
- **Emotiv BrainViz** – for visualising and interpreting the frequency bands and cognitive states.
- **Emotiv BCI** – training the control outputs from specific user-intended brain activities.

BCI Testing Environment

In typical academic or clinical research, EEG signal acquisition is performed in controlled environments such as Faraday cages or signal-shielded rooms to reduce environmental interference and improve signal-to-noise ratio. "To ensure good quality digital EEG recording in clinical use, the following standards have been adopted for recording, storing, reviewing and exchanging EEGs among clinicians and laboratories." [Nuwer et al., 1998] However, we intentionally opted not to replicate this setup. Our goal was to evaluate how robust the system would be in a typical home setting, which is the intended real-world deployment environment for a rehabilitative BCI device.

Even though clinical testing environments are common [Alexander, J. Casson, 2019], we designed our protocol around home use cases. A realistic home setup might include a TV, Wi-Fi routers, phones, and other electronics that can generate electromagnetic interference. We simulated this by running tests in a room with a desktop PC and two active mobile phones all connected to the internet and Bluetooth earphones connect to one of the mobile phones. This

allowed us to understand how resilient the system was to interference in uncontrolled settings.

Environmental Risk – EEG Noise

EEG systems are highly sensitive to electrical and motion-based artefacts [Alexander, J. Casson, 2019]. In our case, common sources of interference included Bluetooth traffic, inconsistent electrode contact due to user movement, and RF noise from mobile devices. Despite the lack of isolation, signal quality was generally adequate due to the real-time diagnostics provided by Emotiv Pro. This allowed us to adjust electrode positions and verify signal quality before each training or testing session. The system's usability in this setting supports its suitability for unsupervised home rehabilitation tasks, provided that best practices (e.g. minimal movement, proper contact) are observed.

Emotiv Pro

Emotiv Pro is useful for examining the fundamental interaction between thought and what the BCI can see. It offers real-time visualisation and signal diagnostics across the 14 EEG channels on the Epoch X.

- Channel Monitoring: The software displays each of the 14 electrode channels as a graph. These channels are mapped to the international 10-20 system for EEG electrode placement:
  - AF3 and AF4: Located on the anterior frontal lobe, monitoring prefrontal cortex activity.
  - F3 and F4: Positioned on the left and right frontal lobes, linked to higher-level cognitive function.
  - F7 and F8: Situated on the lateral frontal areas, associated with emotional and decision-making processes.
  - FC5 and FC6: Frontocentral regions, bridging motor planning and cognitive areas.
  - T7 and T8: Placed above the ears over the temporal lobes, often monitoring auditory and language processing.
  - P7 and P8: Over the parietal lobes, involved in sensory integration and spatial awareness.
  - O1 and O2: Located at the back of the head over the occipital lobe, monitoring visual processing.
- Best Practices: Before any training or data collection session, ensuring that all electrodes are in complete contact and have high EEG quality and signal integrity is crucial. This requires the correct placement and properly soaked saline electrodes to maintain consistently.
- Real-Time Feedback: Emotiv Pro provides a continuous stream of raw EEG signals, allowing users and observers to visualise neural activity across various frequency bands and per electrode.

The array of electrodes allows us to see the different mental responses to stimuli. Everyone has a unique, fingerprint-like response to commands or prompts, meaning that the data can look very different from person to person.
For comparison, here are some different stimulus responses to discuss.

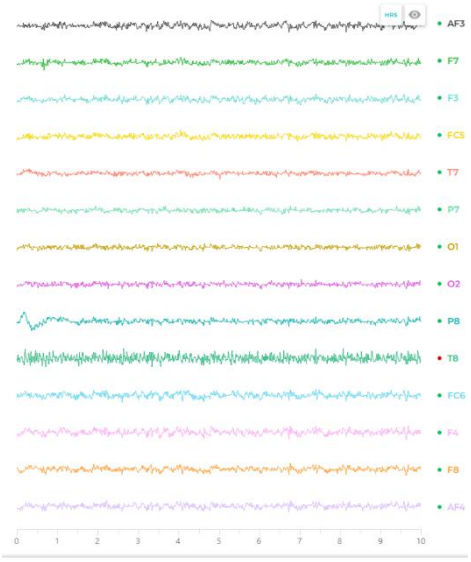


Figure 6 - Push Command

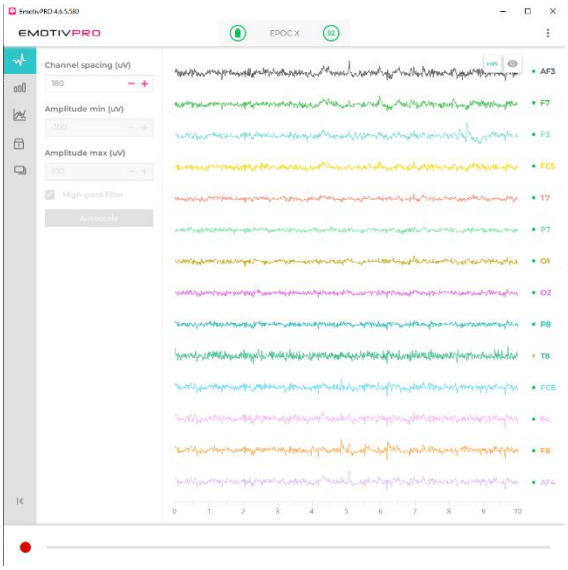


Figure 7 - Arithmetic Operations

When analysing each set of EEG graphs, it's possible to determine which brain regions exhibit electrical activity in response to specific stimuli or commands. The examples provided show that dominant reactions are consistently concentrated in the brain's frontal regions. This includes the prefrontal and frontal lobes, which are commonly associated with executive function, decision-making, and working memory.

Notably, the activity remains frontal mainly even in response to commands like “push, " which one might intuitively expect to engage primarily the motor cortex located further back in the brain. This suggests that, for this user, cognitive processing and intention formulation play a more significant role than direct motor planning during interaction with the BCI system. It outlines how mental effort and focused thought, rather than sole physical motion intent alone, can represent the same command or idea within different people.

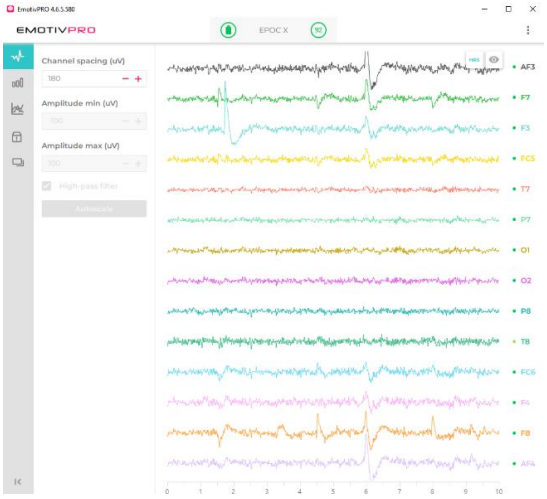


Figure 9 - Recall of Music

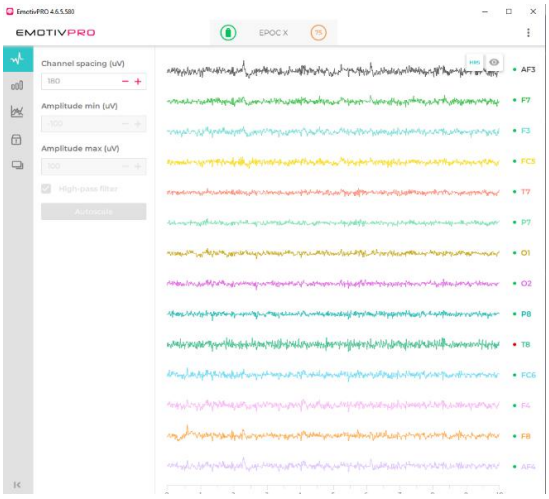


Figure 8 - Animal Recall

Emotiv BrainViz

BrainViz offers intuitive visualisation of EEG data across different brainwave frequency bands. This is particularly useful for understanding user cognitive states and identifying patterns during training sessions.

- **Wave Types**:
    - **Delta (0.5–4 Hz)**: Associated with deep sleep.
    - **Theta (4–8 Hz)**: Reflects drowsiness, meditation, or frustration.
    - **Alpha (8–12 Hz)**: Indicates relaxed wakefulness, calmness.
    - **Beta (12–30 Hz)**: Linked to concentration and mental activity.
    - **Gamma (30–100 Hz)**: Connected to learning, memory, and sensory perception.

These bands are extracted from the 12 primary EEG channels displayed in Emotiv Pro. Understanding these patterns helps in correlating cognitive states with command accuracy during robotic control.

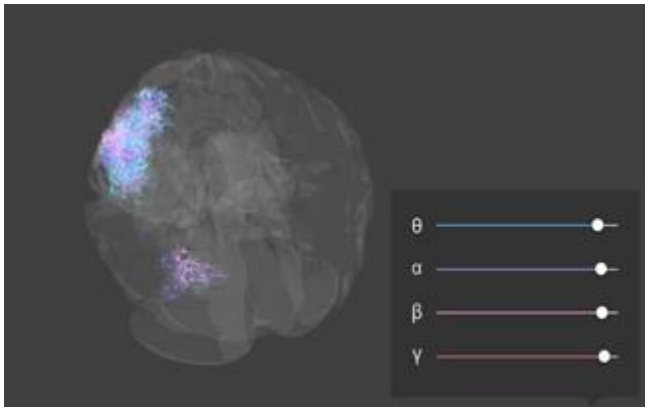


Figure 11 - BrainViz Push Command

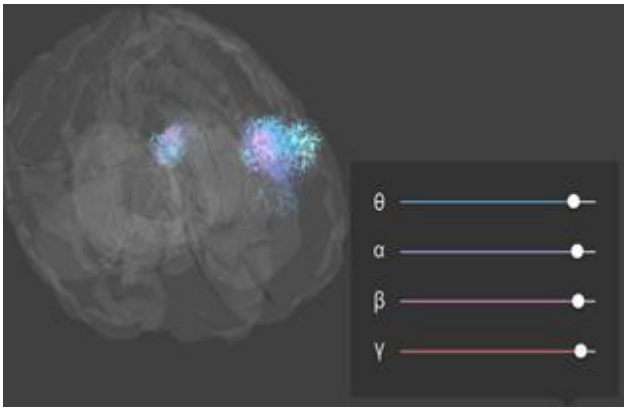


Figure 10 - BrainViz Arithmetic Operations

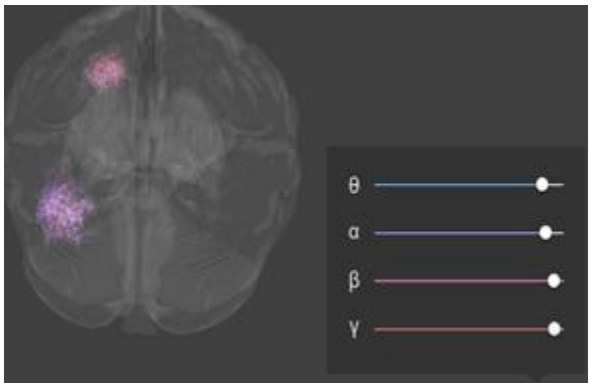


Figure 12 - BrainViz - Animal Recall

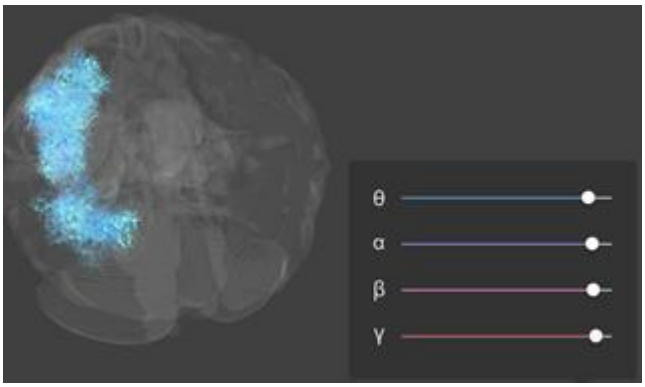


Figure 13 - BrainViz Recall of Music

The variation in brainwave activity across the four figures highlights how different mental strategies used to trigger the same "push" command yield distinct neural patterns:

- **Figure 1 (Simple Arithmetic)**: This strategy activates **theta, alpha, and beta** waves. Theta reflects focused internal attention, alpha indicates a relaxed but alert mental state, and beta suggests active cognitive processing. This combination is typical of structured mental tasks involving sequential thinking and logic.
- **Figure 2 (Mental Push Against Wall)**: The "pure" push imagery shows **theta and alpha** dominance, suggesting that motor imagery taps into subconscious focus and visualisation, with minimal high-frequency cognitive engagement. This is likely the most intuitive and direct representation of the intended BCI command.
- **Figure 3 (Listening to Music Internally)**: This produced primarily **theta** activity. The lower-frequency dominance suggests a passive, imaginative or emotional state, which, while internally engaging, may be less effective for precise command execution due to limited cognitive focus.
- **Figure 4 (Recalling Animal List)**: This figure exhibits **theta and gamma** activity. Theta indicates memory recall, while gamma is associated with higher-order processing and memory integration. This suggests deeper cognitive retrieval processes, which may lead to inconsistent training outcomes due to variability in memory load.

These results demonstrate that **simpler, focused mental imagery (Figure 2)** produces more consistent EEG patterns for BCI training. In contrast, abstract **or memory-based strategies (Figures 3 and 4)** can introduce noise or cognitive overhead. The main issue would be differentiating the mental imagery, which would need further testing. However, this does suggest that training protocols should prioritise direct, visual-motor imagery tasks to improve classifier reliability and reduce overfitting or signal ambiguity.

Emotiv BCI

Emotiv BCI enables the training of specific user mental commands and maps them to actions on the robotic system. The software provides visual feedback on performance, model confidence, and training quality.

- **Training Process**:
    - Users train specific thought patterns ("push", "pull", "lift" and “rotate right”) over multiple short sessions.
    - Visual indicators assist in identifying well-trained models vs. over-trained or poisoned models.
- **Dataset Quality**:
    - Acquiring a good dataset created risks and problems we couldn’t have anticipated. Getting a good dataset requires a multitude of different brain activities; however, since we’re limited to 14 channels, it posed a few issues.

- **Good training data**: Clear and consistent mental effort.
  Here is an example of the best training set we got. Clear, consistent, and range data prevent commands from getting mixed up when using the headset.
  The Push command was a mental visual image of the patient pushing against a wall. This command was the easiest to maintain
  The Pull command was simple, quick arithmetic, yet hard to maintain. If you paused for too long, the command would drop.
  The Lift command played/sang a song in the patient's head. On paper, this is a great decision. However, it was a struggle to receive consistent command outputs, but it was still reliable enough for the task.
  Finally, the Rotate Right command recalls a list of animals. This command seemed to get the least response from the headset and was challenging to output.

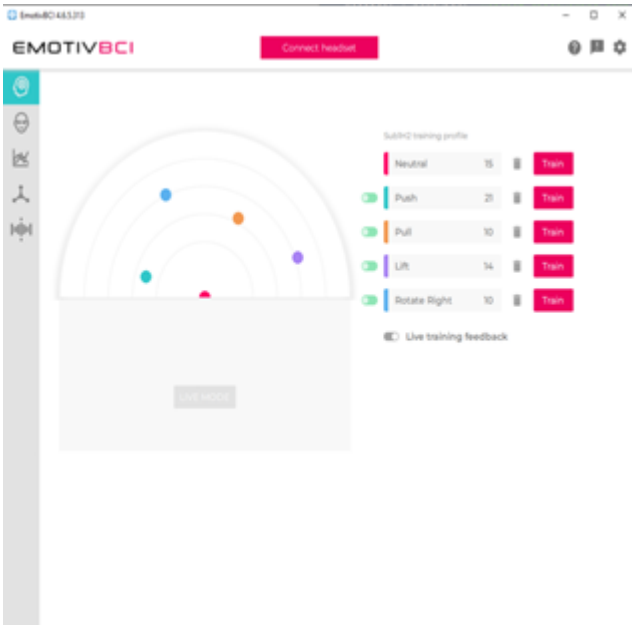


Figure 14 - Initial Training with 5 commands

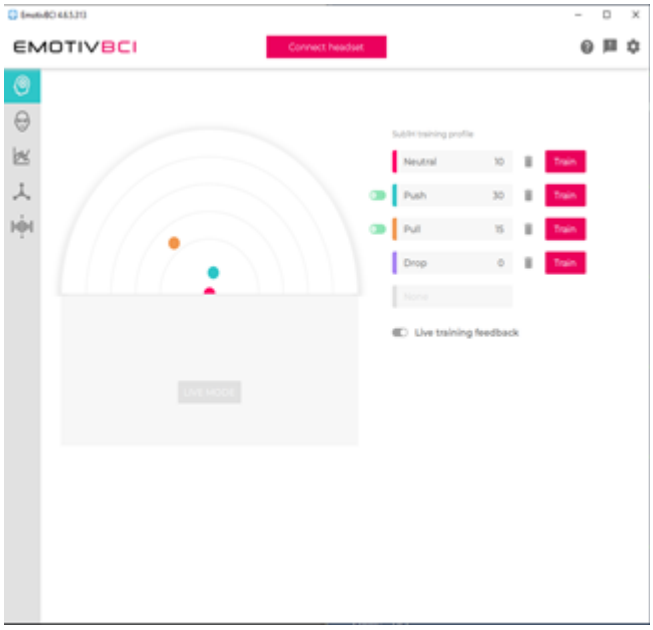


Figure 15 - Initial Training with "Drop" vs "Lift"

- **Poisoned data**: This can be introduced unintentionally by distractions or inconsistent thoughts. This was originally a problem when the central patient tried to sing/play a song in their head. The patient would also view the music video, which would cross-contaminate with the Push command. See the image for an example of the closely related data.
  Additionally, there should always be two people attempting to record data: a patient and an operator. Below is a poisoned data set where the patient accidentally accepted a 0/100 training piece. Since Emotiv is locked down and doesn't allow you to undo or revert data, the entire data set was poisoned, which incidentally contaminates ALL of the data.

Figure 16 - Full Command Set

- **Over-trained models**: This was also an unforeseen risk of overtraining. The best spots seem to be between 10 and 20 sets. Anything over 25, you risk overtraining the command, and it can eventually become poisoned or useless. As stated above, the team's primary patient began visualising a music video with the song they were playing in their head, and this began to bring the data points on the graph too close together and became very hard for the patient to distinguish between the two commands. In this application, you don't want to be mixing commands.

- **Best Practices**:
  - Stop training if concentration wanes.
  - Avoid creating too many actions at once.
  - Avoid overtraining a specific dataset.
  - Only proceed once the system gives high consistency scores for a command.

When observing someone during training, although a user may feel fine, you can often notice mannerisms showing fatigue or stress, which should be brought up to the user to help follow the best practices mentioned above. A user may not at first realise when they are starting to show fatigue, as they may not have learnt what the mental fatigue from thought and BCI training feels like if they haven't used a similar system before.

These changes can be evidenced through tone of voice, changes in breathing – e.g. sighs or heavier breathing – or facial expression, with fatigue often showing as sunken features on the face or more vacant eyes.

Stress can affect not only concentration, but also data quality, as their state of mind differs from when the command was first recorded, or their brain can begin to wander. Fatigue can also reduce data quality. The brain may not create as vivid an image or thought as the original, leading to worse quality being fed into the already unforgiving algorithm.

Hardware

This section details the physical components used to build the system, including electronic modules, mechanical parts and structural materials. Each item was selected to balance functionality, cost and ease of integration within a wearable exoskeleton.

Mechanical

This section outlines the mechanical design of the exoskeleton, including 3d-printed components, joint mechanisms, and fastening methods. The design prioritises comfort, durability, and ease of assembly for rehabilitation use.

Approach

The exoskeleton design process began with creating the bicep brace, which was a critical starting point as it serves as the primary interface with the body. The brace provided a fixed point from which the rest of the system could be developed and expanded, with placement and adjustment of additional elements occurring around it.

We opted for a modular approach for the brace using two C-shaped cups, similar to crutch designs. This choice was practical because it allowed for interchangeable components, enabling easy modifications and iterations without reprinting the entire bicep section completely. This flexibility ensured the system could be tailored to different users, as parts could be resized or adjusted to fit varying arm sizes. Furthermore, straps allowed additional size variation within the same-sized cups, adding to the system's adaptability.

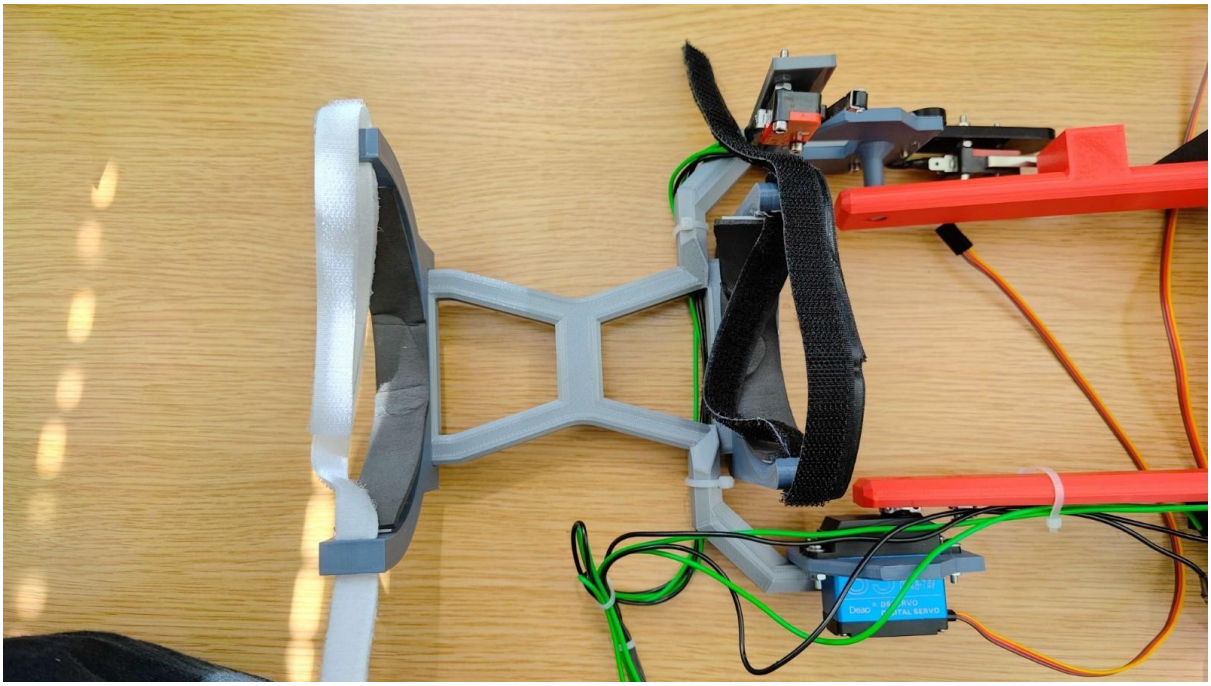


Figure 17 - Bicep Arm Adaptability

The design process included measurements of Harvey to ensure each component fit him properly without being excessively bulky. Since the system was intended for prototyping and rehabilitation, it was essential to maintain a usable, if unrefined, form factor that could be worn during testing and demonstrations.

Working within the constraints of available equipment and components, we made a conscious decision to prioritize the system's technical feasibility instead of achieving full mobility. More advanced manufacturing methods and materials could theoretically enable smaller, more substantial parts, but given our setup, we focused on achieving functional reliability using accessible means. PLA was chosen as the primary material due to its fast print times and suitability for rapid prototyping. This allowed us to quickly iterate small components, especially, and iterate design elements to fit between, even if PLA's strength and durability fell short of theoretical alternatives.

Once the bicep brace was established, the next critical step was determining the optimal placement of the servo on our test subject, Harvey. Ensuring that the servo allowed proper elbow flexion was essential for testing the effectiveness of the mechanism. We positioned the servo at the elbow crease. This area exhibits minimal movement when the arm is straight, ensuring the system does not hinder the elbow's natural motion.

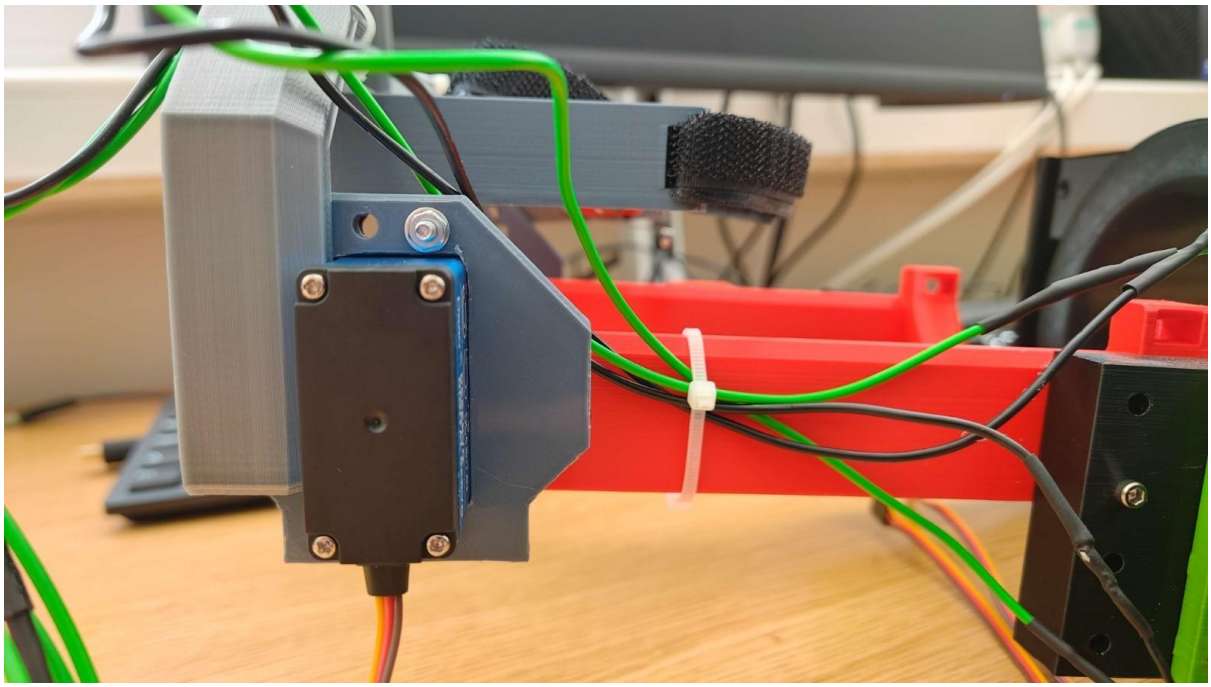


Figure 18 - Elbow Servo Motor

With these foundational elements in place, we focused on designing the system to allow for iterative development and easy modification. A key factor in this approach was the limitations imposed by our 3d printing setup, particularly the Ultimaker 3. This printer had a fixed build volume, which set a size constraint for the parts. However, it was crucial to maximise the length of components that would extend along the arms. More significant parts reduced the fastening points, decreasing potential failure points at these critical joints.

Another important consideration was the need for frequent iteration. Design changes were inevitable as we adapted to new insights, unforeseen challenges, and evolving safety and

performance requirements. This iterative process allowed us to refine and improve the system continuously.

The design also required careful consideration of part separations. Making each component a separate print was necessary for several reasons, including accommodating the printer's build volume and allowing for individual modifications. This made the process more flexible and responsive to design changes, while reducing the complexity of reprinting entire sections of the system for small quality-of-life or mechanical tweaks.

The design elements of the exoskeleton were broken down into various parts, each designed to serve a specific function within the system or to support a safety mechanism. Key components included:

- Bicep brace
- Arm cups
- Elbow servo mount
- Elbow non-powered pivot
- Elbow upper and lower limit switch mounts
- Forearm brace
- Wrist collar lower mount
- Wrist collar upper mounts
- Wrist mount side bars
- Wrist collar
- Wrist servo mount
- Wrist limit switch mount
- Wrist linkage bar

Each component was individually printed, paying attention to print orientation and infill density and balancing strength, material, and time efficiency. This ensured that the parts were durable where necessary for safety and function, while minimising the downtime between iterations that could complicate the printing process for future additions. Additionally, the modularity of the design allowed for easy repair when things did unfortunately go wrong, with individual reprints keeping downtime minimal and helping to maintain momentum during development and testing.

Designing individual parts was simple yet time-consuming, as one had to consider both the size and spacing of arts relative to where the design could be foreseen and maintain an awareness of hole placement for heat-sunk threaded inserts. Most construction uses M3 bolts, threaded inserts, and nuts throughout the design. This made assembly easy and meant that designs were consistent and did not require multiple tools to assemble and disassemble. Only a hex key and a screwdriver are needed to turn bolts, and pliers or sockets can hold nuts in place.

Designs also had to be designed with the awareness that certain elements needed support to maintain quality. Although some designs could be printed, adding extra design elements, especially on the exposed wrist servo bracket, was necessary to prevent damage in axes that do not align with the layer lines' optimal strength.

Mechanical Analysis of Parts

To save printing time, some analysis was done on the parts to check that they wouldn't buckle under the pressure from the user's arm or the force exerted by the servos. This was prioritised on the parts expected to exert the most stress on them. In contrast, others were deliberately designed to have an extremely high safety factor, preventing the need for analysis on all parts. The high stress parts were often in contact with the servos or around the elbow or wrist joints, as these have high momentary or single point forces.

The parts all use PLA (Polylactic Acid), which can be printed with different infills. Because of this, each part could be modelled with different infills, resulting in varied results. At 100% infill, the Young's Modulus was taken as ~3.2 GPa. At 50%, it was taken as ~1.25 GPa. At 20%, it was taken as ~0.3 GPa. This shows a dramatic shift in the material strength between infill levels, which could result in very different performances from the different parts. It also means that while the lower infill could have been used for the less stress-intensive parts, an infill that was too low could also result in those parts failing.

In addition, the average mass of the human arm is ~5% (reference: the human machine) of the weight of the entire human body. The subject who tested the arm had a mass of close to 4 kg. This meant that stress and strain calculations based on the initial values given were possible.

A factor of safety of two is standard for a project of this nature so that the parts can withstand any extreme scenario that may be unforeseen in the mechanical analysis. While the expected loads may be lower, once the prototype is operational, environmental factors may align to produce forces higher than anticipated during the design phase.

Rigid Link:

The rigid link required the most trial and error, as it was only a small part compared to the rest of the arm. This meant that parts could be fabricated and broken much quicker than any of the others.

The rigid link is the part which connects the wrist-mounted servo to the bearing on the inside of the arm, which translates the movement between the two. Because of this, it needs to be able to withstand high shearing stresses, particularly while still being a small and thin part to fit within the confines of the parts surrounding it.

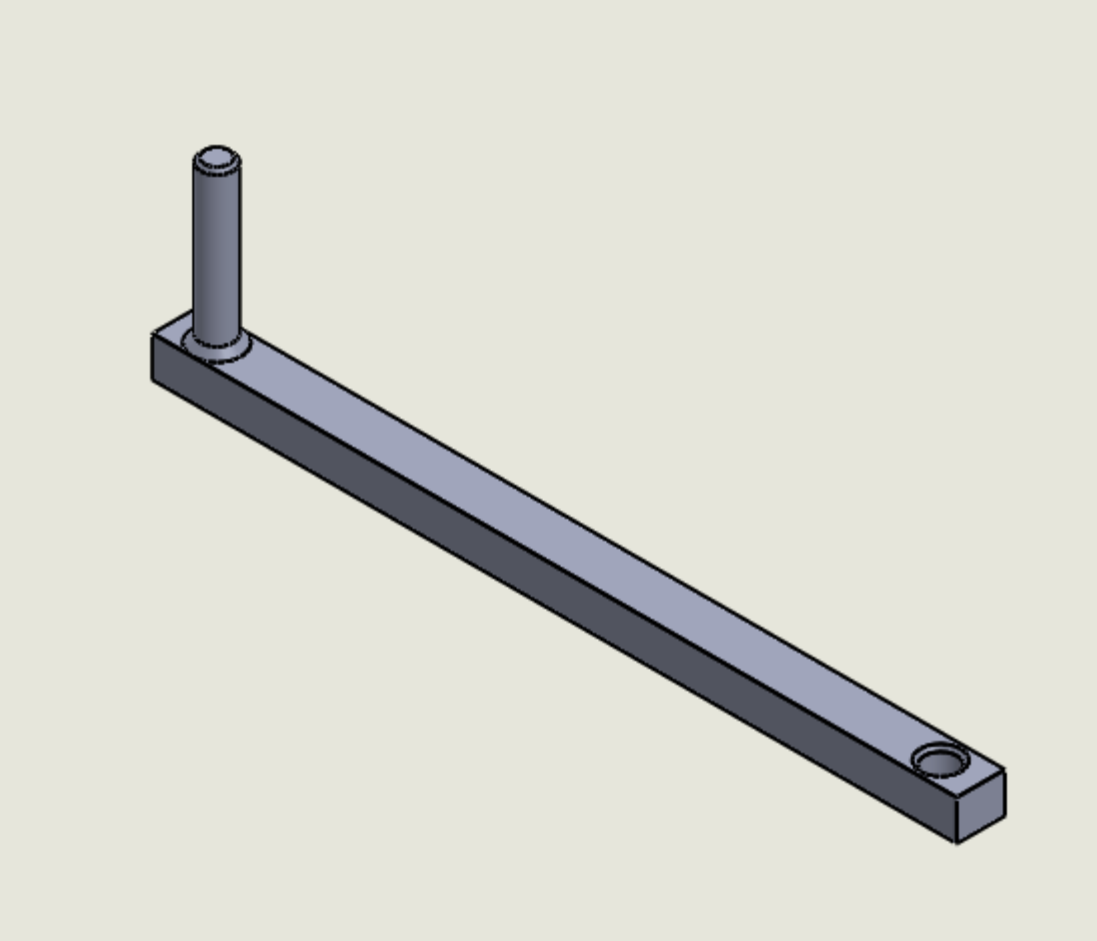


Figure 19 - Wrist Linkage Bar

The cross-section of the part across the rigid link is a 5x5mm square which means that the flexural stress and strain should follow a parabolic distribution across the height of the square. The Equation is as follows:

$$\sigma = \frac{My}{I}$$

Where:

σ = bending (flexural) stress (MPa)

M = applied bending moment (Nmm)

y = distance from the neutral axis to the point of interest (mm)

I = Second moment of area (mm^4)

For a rectangular bar:

$$I = \frac{bh^3}{12}$$

And:

$$y = \frac{h}{2}$$

So, in this case:

I = 52.08mm^4

y = 2.5mm

σ = 68MPa

Solve for maximum moment:

$$M_{max} = \frac{68 \cdot 52.08}{2.5} \approx 1416N \cdot mm$$

Any moment above this value would result in the part breaking.

Next is to find the breaking load:

P=4MmaxLP=4MmaxL

Where:

P is the breaking load

L is the length of the beam

P=4×1416100=56.64NP=4×1416100=56.64N

This value is above the expected load, so the part is adequate for the system. This showed that 85% infill would be adequate for the part's usage.

The everyday stresses on this part aren't an issue, as the high-density infill results in the part being resistant to normal loads. If the infill had been lowered, this may have resulted in concerns. However, all issues with this part were related to its bending down its longest side. Also, it will always fail due to the bending stress before any other method, such as buckling or axial stress. There was also no rotational force put onto this part, meaning that shear and other rotational stresses didn't have to be considered.

Bicep Upper & Lower:

The parts for the bicep were adapted for the comfort and ease of use while still being able to hold the full weight of the arm. While it wasn't likely that the entire weight of someone's arm would be resting on one of the pieces, this was considered an extreme circumstance, with factors of safety being considered as well. The area for these parts was more complex to calculate due to their distinct "C-shaped" design. In addition, we decided to assume that the dispersion of the forces is uniform across the surface or the "horseshoe" element of the part. In contrast, the actual forces will be higher at the centre and less concentrated at the edges due to the nature of the design.

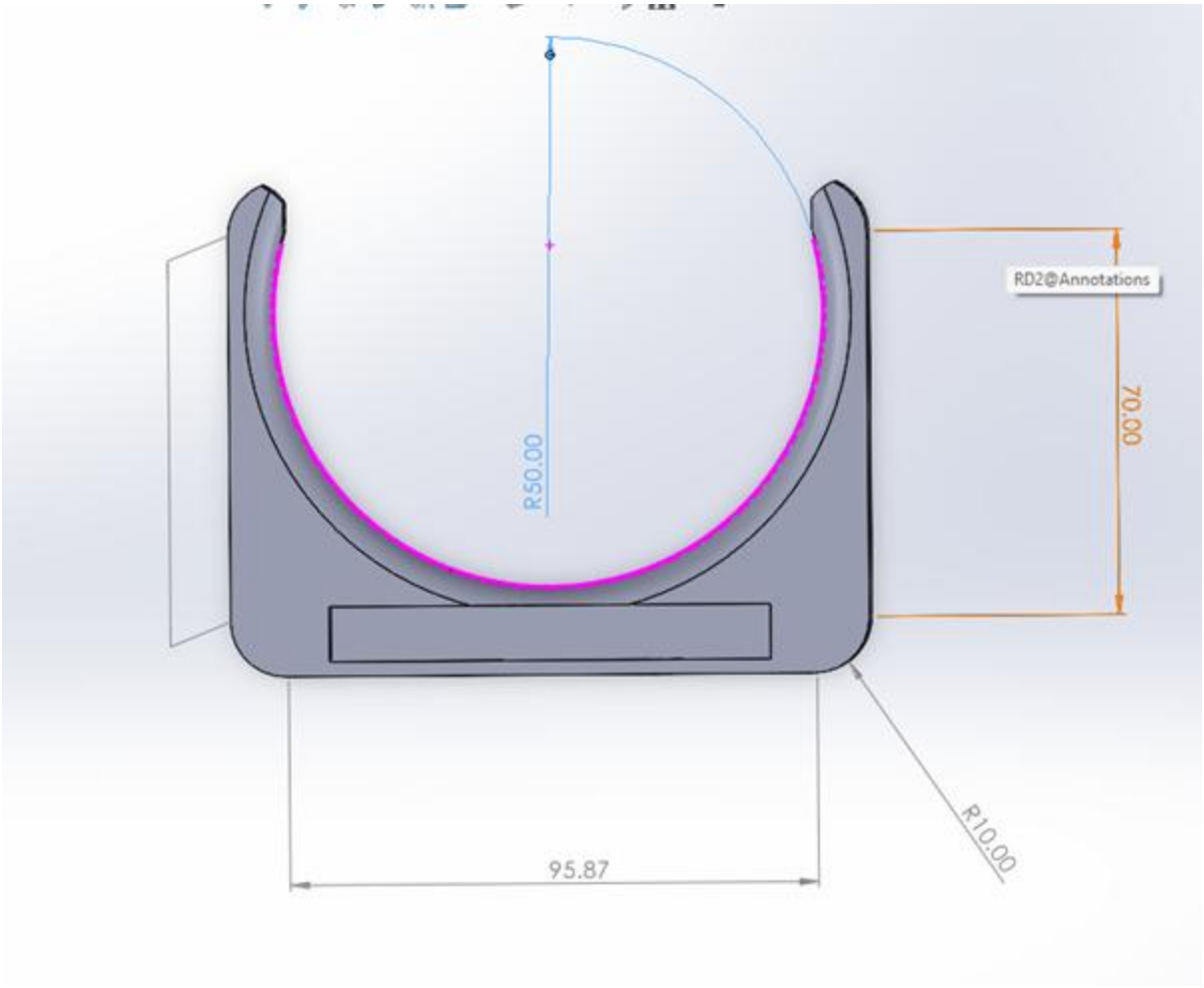


Figure 20 - Bicep Clamp

The upper Bicep is a broader part to allow the upper bicep to fit through it quickly. As well as the measurements shown in the figure above, the part has a depth of 20mm, providing a suitable surface area for the arm to rest on. The highest possible stress can be represented by a rectangular load of 4 kg (39.2 N) across the thinnest point of the structure, which is 20mm thick. We assume that the contact parallel to the arm only represents 5mm, which could be an underestimation for safety reasons, as a more spread load could lead to a false safe result. This also represents the weakest point of the lower bicep piece, which has a very similar design. This was testing the parts at 50% infill to see if print time can be reduced as well as material usage without ramifications on the part stability.

Calculate the Area where the force is acting:

$$A = 20mm \times 5mm = 100mm$$

Work out compressive stress:

$$\sigma = \frac{F}{A} = \frac{39.2N}{100mm^2} = 0.392MPa$$

Work out the max compressive strength for 50% infill (Given compressive strength of PLA at 100% infill is 65 GPa):

$$\sigma_{max} \approx 0.5 \times 65MPa = 32.5MPa$$

Therefore, the predicted compressive strength for the part is far below the material's breaking point and can thus be printed at lower infill densities. Also, considering that it is unlikely for the entire weight of the arm to rest on one-part, large cutbacks can be made from this and other parts of the arm.

Elbow Plates

The elbow plates proved challenging from a design standpoint as they considered a wide range of stresses during operation. The primary stress that had to be considered for this part was bending stress, as there needed to be limited contact points across the elbow to facilitate movement from the servo. This, however, causes high point loads at those contact points, resulting in high momentary forces across the part. Because of this, it was immediately apparent that this would have to be a higher infill part due to the bending stress alone.

Another stress to consider is the axial stress caused by the lower arm wanting to pull the structure away from the bicep section of the system. This had less of an effect on the structural integrity than the bending stress. However, it was still higher than initially expected. There is also a high amount of shear stress going through the part's cross section due to the servo's rotating motion and the bicep parts wanting to pull the part up while the lower arm is acting with gravity to pull the structure down.

Strain should also be considered for this part, as it could cause it to deform over time, leading to a change in shape or potential breakdown later on.

This is the part that would have to be replaced the most frequently in real-world use, as it is essential to the structure itself while also facilitating movement, which will wear the part more quickly than the stationary parts found elsewhere in the system.

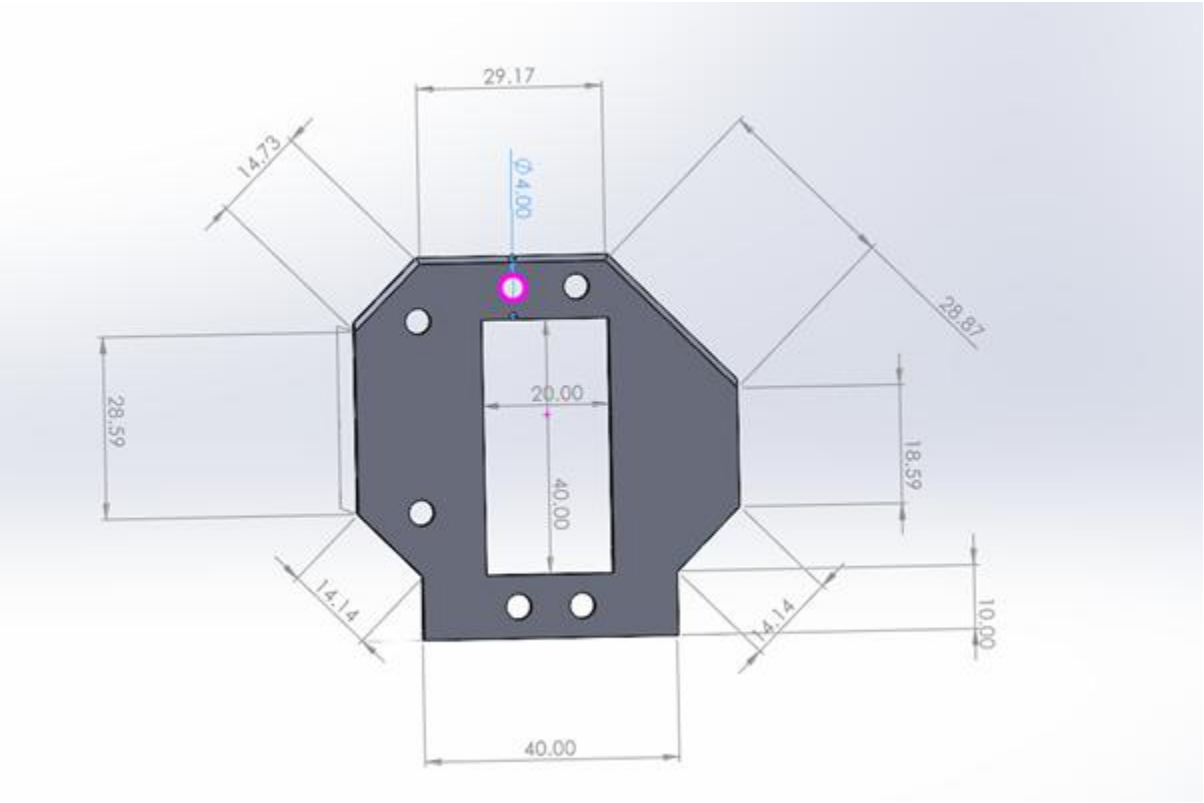


Figure 21 - Elbow Plate

The part is designed to be broad with multiple points of contact on each connection to ensure that there isn't an unreasonable amount of wear on a particular connection. The depth of the part is 5mm, meaning there could be potential for tension to cause bending in high-stress areas. This particular connection is holding the servo, which is articulating the elbow joint, which is held in the gap left in the middle of the part, which may cause it to be less prone to breaking than the other part due to weight distribution.

Finding the Bending Stress of the Elbow Plate:

$$\sigma_{bending} = \frac{My}{I}$$

Where:

M = bending moment

y = distance from neutral axis

I = second moment of area

Calculate the bending moments (when taking the neutral point as 10mm from the centre of the connections, assuming half an arm's weight is on each connection) (weight of a 35Kgcm servo is 3.4N):

$$M_1 = 19.6 \times 10 = 196N \cdot mm$$

$$M_2 = 23 \times 10 = 230N \cdot mm$$

Calculate the moment of Inertia for the rectangular section:

$$I = \frac{bh^3}{12} = \frac{5 \times 59^3}{12} = 85574mm^4$$

Calculate the bending stresses:

$$\sigma_1 = \frac{196 \cdot 10}{85574} = 0.02MPa$$

$$\sigma_2 = \frac{230 \cdot 10}{85574} = 0.03MPa$$

Therefore, the elbow plate's design has significantly reduced the likelihood of it being bent to the point of deforming or breaking. The elongated design in the vertical direction significantly reduced the stress on the cross-sectional area of this part, resulting in the stresses being reduced to near-negligible levels. Also, there are two of these pieces, one on either side of the arm, resulting in even more stability.

Overall, this analysis, whether fully fleshed out or just roughly worked out on paper, allowed us to understand how the parts would behave before printing. In addition, it could warn us about parts that could potentially fail ahead of time, meaning that we could hold back a print, saving time and resources. When print times can potentially exceed multiple days, this was a effective way of testing designs.

Iterations and Problems

Most of the problems arose in the fabrication and assembly of the system.
A common recurring issue was that parts would print with the walls under-extruded and not fully attached to the part, only at the corners. This issue was fixed by changing the temperature of the printer's nozzle and increasing the flow rate. This, along with the most reliable printer, allowed parts to be consistently printed at the desired quality to ensure reliability with the system.

The second major issue was the available hardware. The initial servos were stronger than those used in the final implementation. However, they didn't come without their own quirks and learning experiences. They remembered the final position they were in when they lost power. However, they also assumed they hadn't moved between then and being turned on. This memory then limited the servo to 180 degrees as specified in the data sheet [Appendix D – DS3240 Servo Datasheet] however they also adjusted where the limit of the angles was relative to the angle they thought they were at, not based on a true specified data point or encoder. This meant that if the arm was powered off at full flexion and then turned on flat, when told to go straight, it would power past that point and go to the specified angle as it had no knowledge of the angle being reflexive instead of where we as people wanted it to go. Once we knew this was how the servos worked, we switched to some other ones with limited angles that return to a fixed position for a given angle rather than scaling angles based on where they thought they were from their memory. This made not only designing but also testing to check for elements needing iteration and modification, as positions were then consistent.

After solving these significant issues, iterations came as adapting parts to meet strength criteria to withstand forces, as parts showed flex and deformation when under a load with resistance from the human arm.
These iterations included adding thickness and changing infill of the parts where points of failure were shown to be, mainly where layer lines and fasteners had concentrated the effects of forces acting upon them.
A prime example of this type of iteration was the bicep brace, as the infill around one of the heat-sunk threaded inserts gave way under a load-bearing test using the servo while the device was being worn. This was a test aimed at seeing whether the servo could move the arm as it stood, but the bicep brace flexed where the servo bracket wings were, and the group heard an audible crack. This was fixed within the day as it was an increase of infill from 30% to 50% and the thickness of the parts and number of walls was increased at the points of and around the failure point.

When testing the infill strength of parts, we used the smallest and thinnest part—the wrist linkage bar—to test strength. Having tested an earlier iteration of the bicep brace, it showed no external failure under 10kg of load; however, the linkage bar dubbed "type I" for testing purposes yielded to breaking at 10kg.
For testing we used "types" of bars of "I - VII" and testing discussed in a following section will further explain this; leading to the thought processes of upping infill for parts mentioned prior to be implemented.

Later iterations also included changing the wrist collar to have a slot for a belt, which was quickly dismissed after viability of moving it with such a mechanism was deemed too advanced for the available time with the lack of resources we had. Instead of this, we opted to use the rigid bar and just push and pull the collar within a limited range as can be seen in the final design.

Realisation of the Mechanical Design and Orthographic Views

The exoskeleton was developed as a semi-modular system to bring the design to life. This modularity was central to both the assembly process and future adaptability, allowing parts to be easily replaced, upgraded, or adjusted without reconstructing the entire system. The various sections were designed to connect using standardised mounting holes and were physically connected using M3 fasteners and, where possible, heat-sunk threaded inserts. In areas where mechanical fastening was impractical or felt insufficient, hot glue was selectively used as to provide structural reinforcement or positional stability. Engineering drawings for measurements and parts not discussed explicitly can be found in [Appendix A – Engineering Drawings]

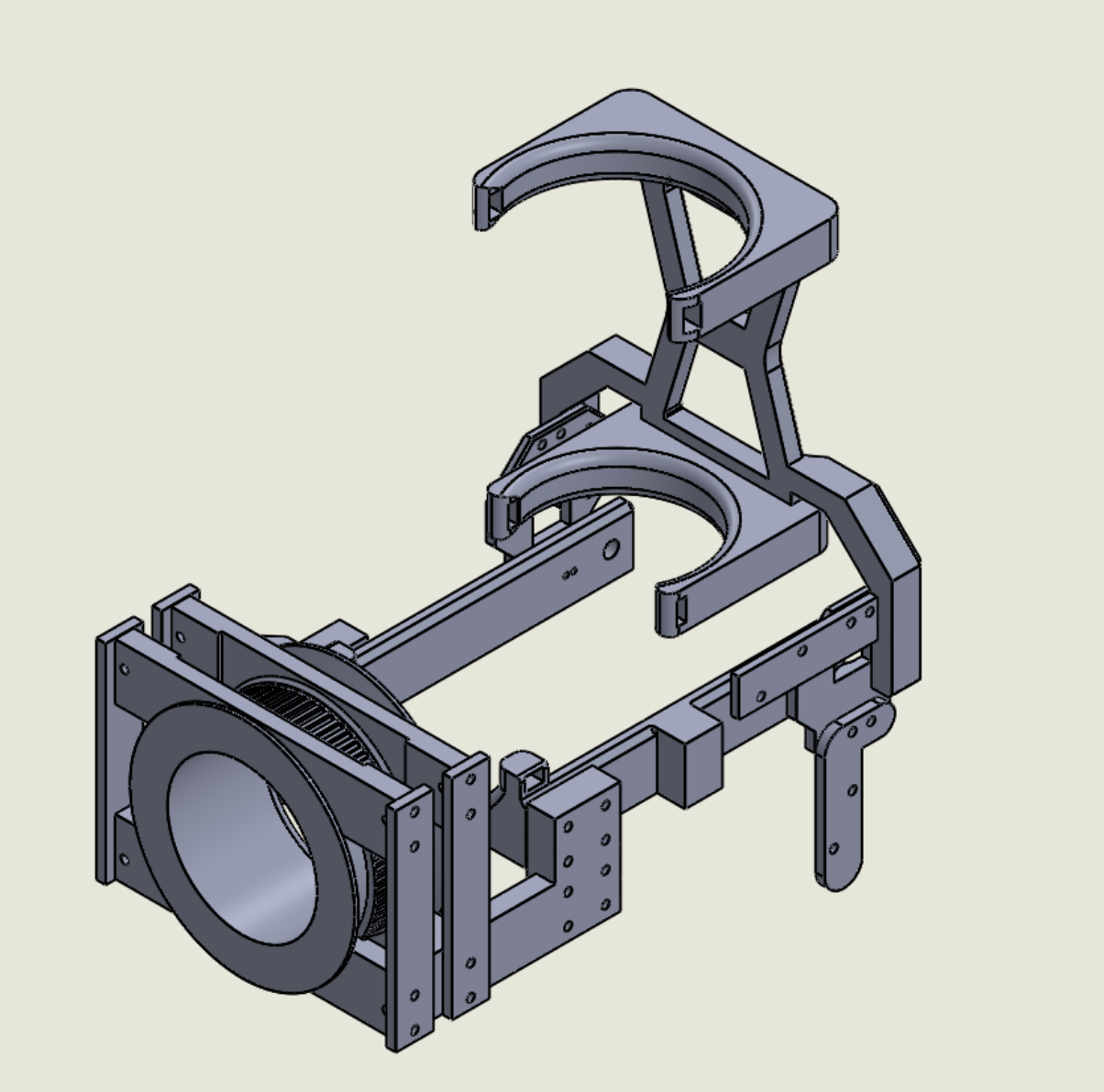


Figure 22 - Isometric view of entire system

The structural elements of the system were entirely fabricated using FDM 3D printing. Components that could not be 3d printed - such as servos and limit switches- were sourced as off-the-shelf hardware. Other non-printed elements, like the interfaces for mounting to a user's arm, were made by hand using craft foam.

Figure 23 - Upper Bicep Cup Foam

This hybrid approach, combining printed and non-printed elements, allowed the system to meet its mechanical and ergonomic requirements while remaining accessible and easily manufacturable with the tools and materials available to the team.

When considering parts such as the forearm brace, printing orientation and making the part as long as possible was important. Despite the necessity of staying within the size limitations, reducing the number of breaks within a section helps reduce possible undesired movement as parts are then connected to the lowest possible number of other parts in a row when facilitating the required length to meet the requirements for forearm length. Having the wrist mechanism on a single large piece and then building off from it also maintains the thought process of keeping minimal parts that shouldn't move separately as one. As the rotation mechanism required iteration, the mounting to the rest of the frame could be tightened to allow consistency when testing each change to find a system that worked reliably. This became especially poignant when testing the limit switch placements, as these required the correct spacing to actuate reliably before rotation too far became an issue without ending travel prematurely.

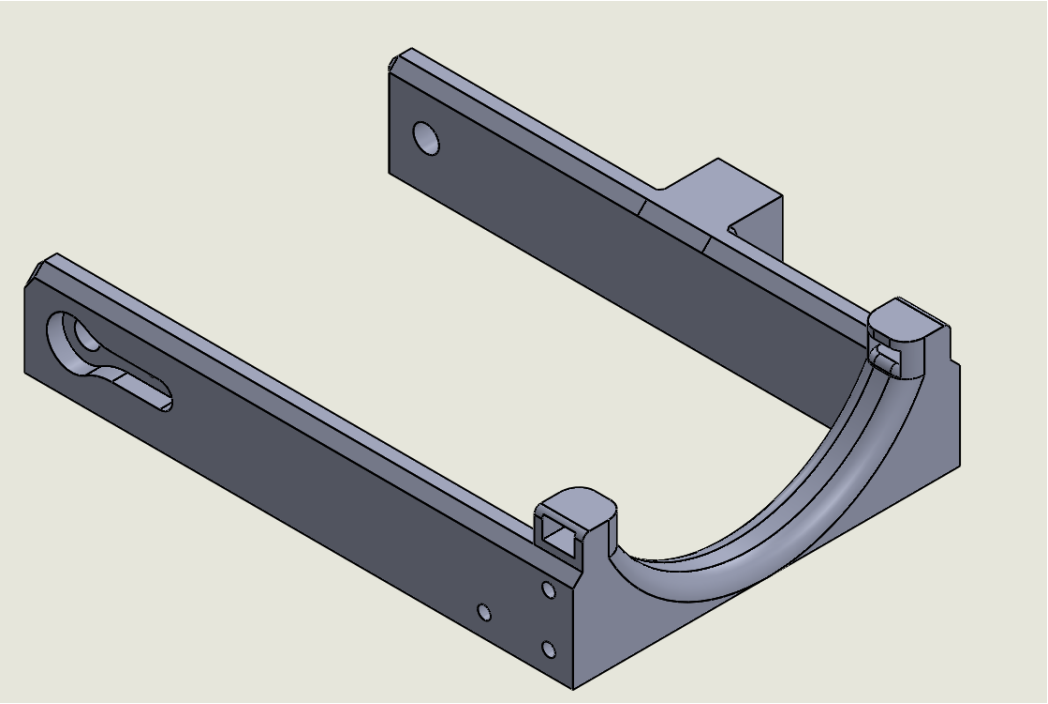


Figure 24 - Isometric View of Forearm Brace

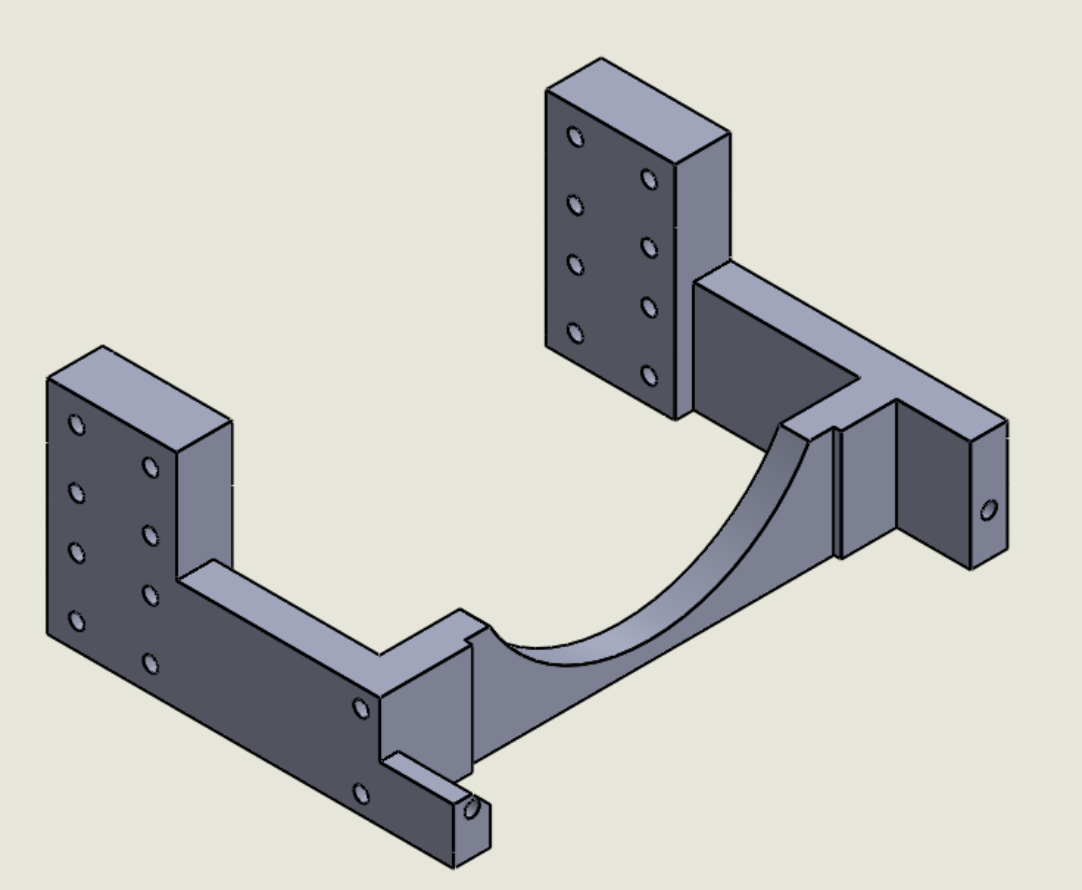


Figure 25 - Isometric View of Wrist Mechanism Mount

Above the elbow was a combination of the bicep cups and the bicep brace. These parts are the system's root, mounting the rest of the system to the user. These parts were printed in differing orientations when compared to each due to their sizing and required use orientation. Since the bicep brace takes a lot of the forces applied due to the user and system, it was important to ensure that the weight of everything that it precedes was not applied perpendicular to layers, as the part is thinnest in its axial direction going down the back of the bicep. Instead, it is pulled parallel to the layer lines, meaning that any axial load would pull along the layer lines, having an increased surface area across each layer to prevent shearing along them.

The cups were printed with the “C” shape flat on the bed. Despite this putting the force exerted by the fasteners under load perpendicular to the layer lines, it allowed the curvature of the cups to be more rounded, as printing these in the same “C” orientation as the forearm brace would make them appear “stepped” due to the layers, especially at the centre of the curve.

Figure 26 - Difference between print layer orientations

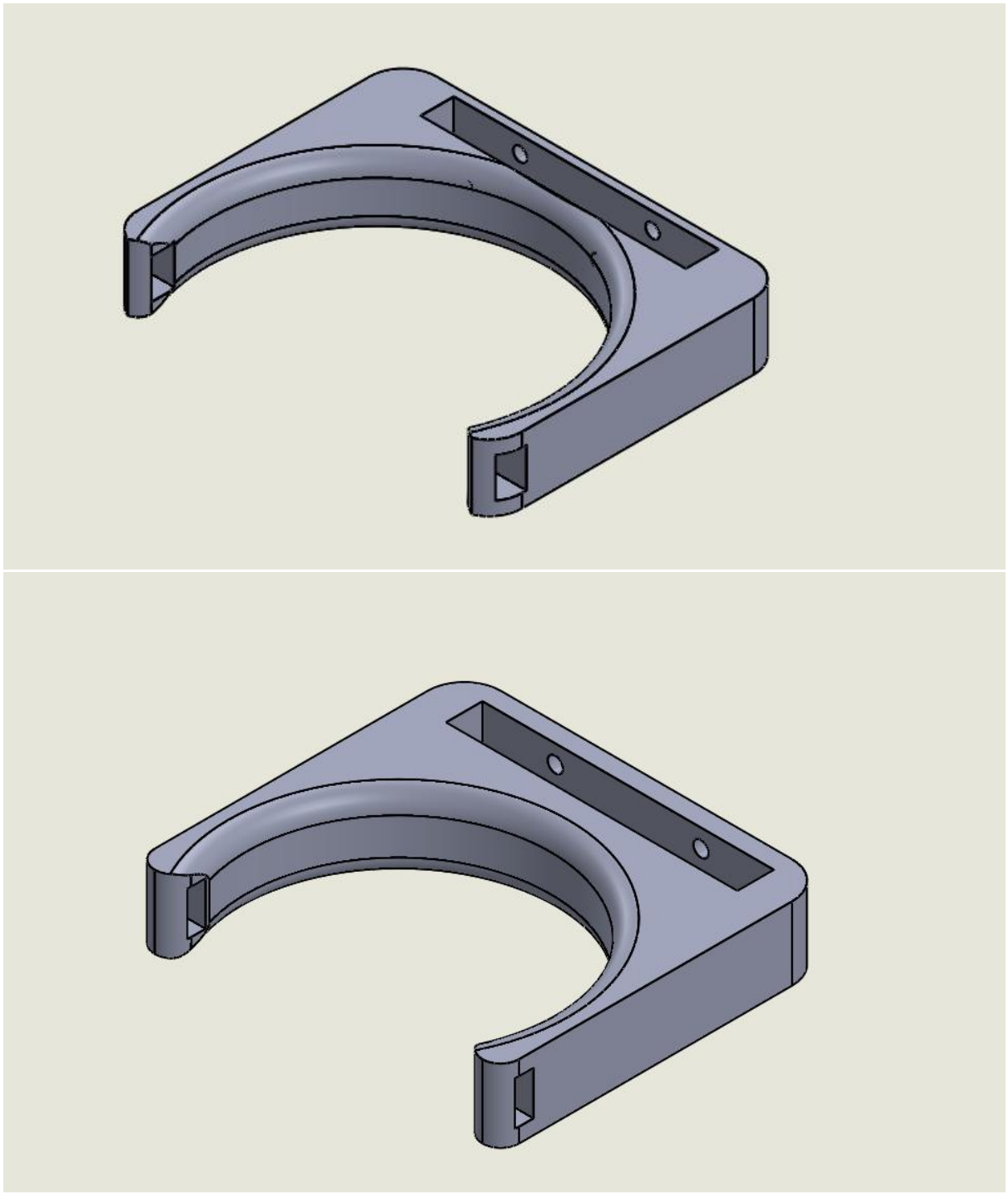


Figure 27 - Bicep Cups Design Iterations

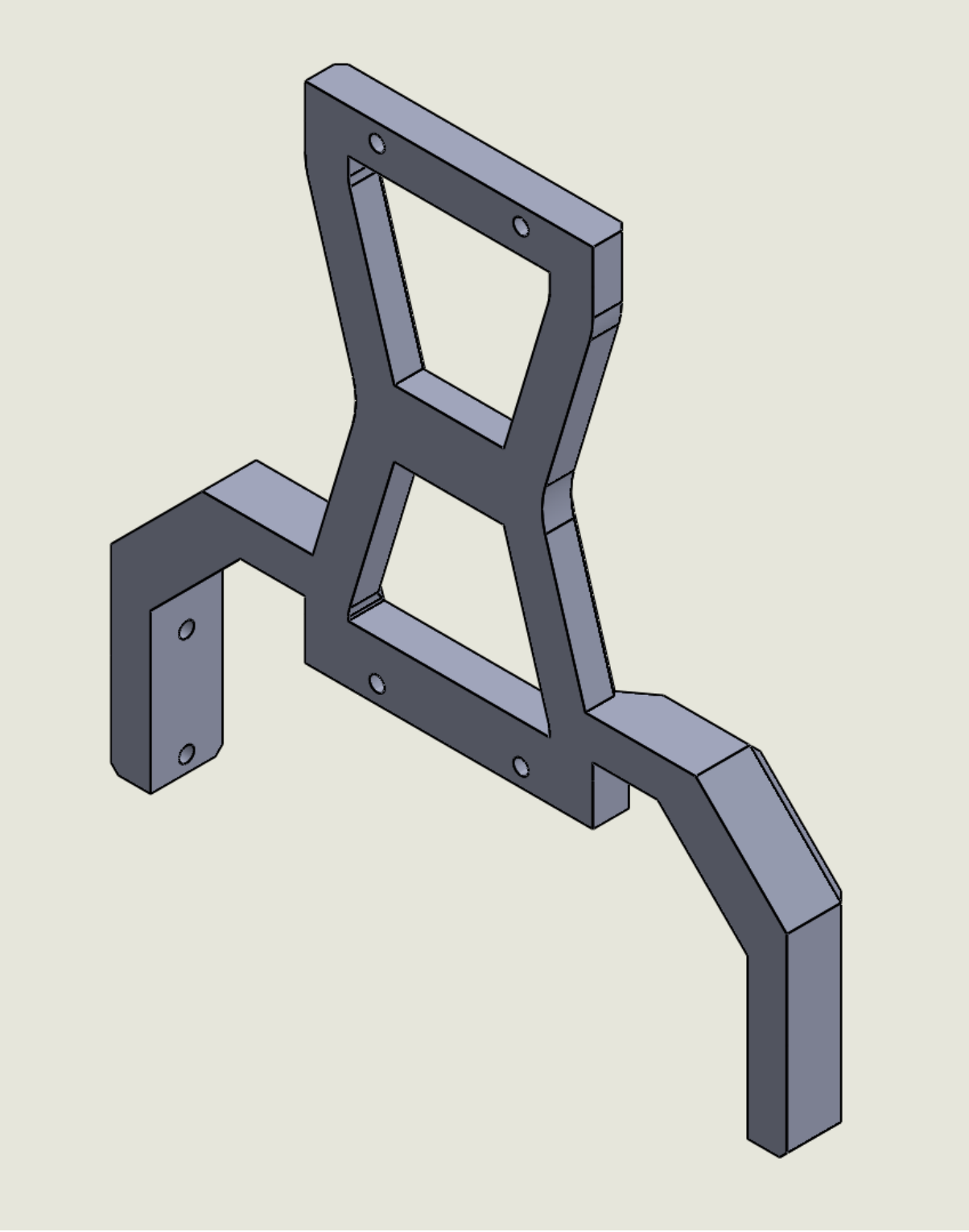


Figure 28 - Isometric View of Bicep Brace

Justification of Final Design

When finalising the design, many final decisions were made, and reasons for the design choices were given.

Firstly, given the testing of various parts by casual, accidental, and deliberate means, the infill of parts between 25% and 50% for any major parts and the infill of the smaller parts ranging even further up to 85% (due to the lack of noticeable benefit from higher percentages) is justifiable for a range of reasons.

It has made the parts durable, and the oldest parts that remain in use on the final construction, even when put under a duration of continuous operation, have withstood the use and testing over time. This was an accidental test caused by a script Leo made for the video to have it move for some of the shots. During this accidental stress test, we found that the shaking for around an hour of use without someone using it to dampen the vibration and moderate changes in angular velocity loosened and, in one case, fully unscrewed nuts and bolts. This is partly because the nuts were not tightened to a fixed and tested torque value and were primarily done by hand with a socket and hex key to be easily disassembled again for repairs or alterations over the development period.

The servo directly interfacing with the elbow actuation allows for simple and effective angular control. However, it also means that the weight felt by it due to a user's arm and the system itself was all on one servo. This was done to keep the design as simple as possible to showcase the technical capability of the over-arching concept of the control mechanism rather than perfect overall operation due to the prototype nature of the project. This same concept of it being a technical showcase is also justifying the wrist rotation servo, only enacting a function of rotating approximately 45 degrees, as it shows the function and control without over-complicating the mechanical aspects of working with an angle-limited servo for a large rotational motion adjacent to its placement to the collar.

The use of curved crutch-style cups padded with foam was decided upon due to the availability of materials facilitated by what could be acquired from our own and the supervisors' available resources. This allowed the parts in contact with body parts to be more comfortable and ergonomic since it isn't hard plastic rubbing on skin or digging in. This is a fast, simple and effective method of accomplishing the minimum requirements for a prototype. However, it allows for future iterations to build upon this if desired, where a better attachment method than hot glue and foam tape could be implemented, possibly with a more comfortable foam and a fabric cover.

Testing

When testing the strength of the parts, a range of dedicated and group tests were conducted, varying in scientific capacity. The most detailed of these was the strength testing of the wrist linkage bars—specifically, the seven different types previously mentioned.

To carry out these tests, we used luggage scales to measure the load applied, which is why the results are recorded in kilograms. Although not ideal, this method was chosen due to the lack of suitable calibrated weights to hang from the parts for a truly scientific test. The scales were pulled manually until the breaking point was reached, introducing some imprecision but still offering usable comparative data.

Initial testing was focused on determining the breaking point of the earliest bar prototype following material test one (M-T01) as described in the M-T series, later referred to as "type I." The peg section, which slots into the wrist collar, consistently failed at or below 1kg of load, a result that repeated across all other variants regardless of changes to infill percentage or pattern, meaning that a concise result was not acquired. This indicated that the peg's inherent geometry, rather than its internal structure, was the primary point of failure, failing above the chamfered base, yet not shearing directly on layer lines as expected. For the main body of the bar, "type I" showed a breaking point of approximately 10kg. This established a rough upper limit for the bar designs and helped to define their mechanical boundaries. However, the sudden snapping of parts during this testing posed a risk to both the tester and the equipment, so later tests avoided complete destructive testing when the expected results were already reasonably well understood.

Subsequent testing employed a measured bend test. Testing was not able to simulate the axial loading the part would experience in use, as we lacked equipment capable of applying a load along the length of the bar and in line with the print layers. Instead, the bending test applied force perpendicular to the layer lines, which is not the primary direction of stress during normal operation.

Due to the essential nature of the test setup and variability in print quality, the data collected should be seen as indicative rather than definitive. Print inconsistencies, manual measurement error, and the non-standard method of load application all contribute to a significant margin of error in the results.

One precise observation from these tests was that for small parts of this kind, variations in infill percentage had a minimal effect on overall strength, at least when using the same internal pattern at high infill densities. This suggests that for certain design elements, especially those prone to higher stress, shape and wall count play a large role in mechanical performance that infill density alone cannot match. A lot of this strength and lack of differences shown will be due to all having the same number of walls, and top and bottom layers.

Each type had the following specifications:

- I - 85% gyroid
- II – 85% cubic
- III – 60% gyroid
- IV – 60% cubic
- V – 60% lines
- VI – 85% gyroid, ½ speed
- VII – 100% gyroid

Table of bending under load in mm				
Type	1kg load	3kg load	5kg load	7kg load
I 85% gyroid	5	14	27	38
	4	15	28	40
II 85% cubic	5	14	24	38
	5	12	25	36
III 60% gyroid	6	20	27	33
	6	22	29	32
IV 60% cubic	6	19	26	37
	7	18	25	36
V 60% lines	7	17	25	40
	7	19	21	42
VI 85% gyroid, ½ speed	4	14	25	36
	4	15	28	39
VII 100% gyroid	3	14	26	35
	4	14	27	33

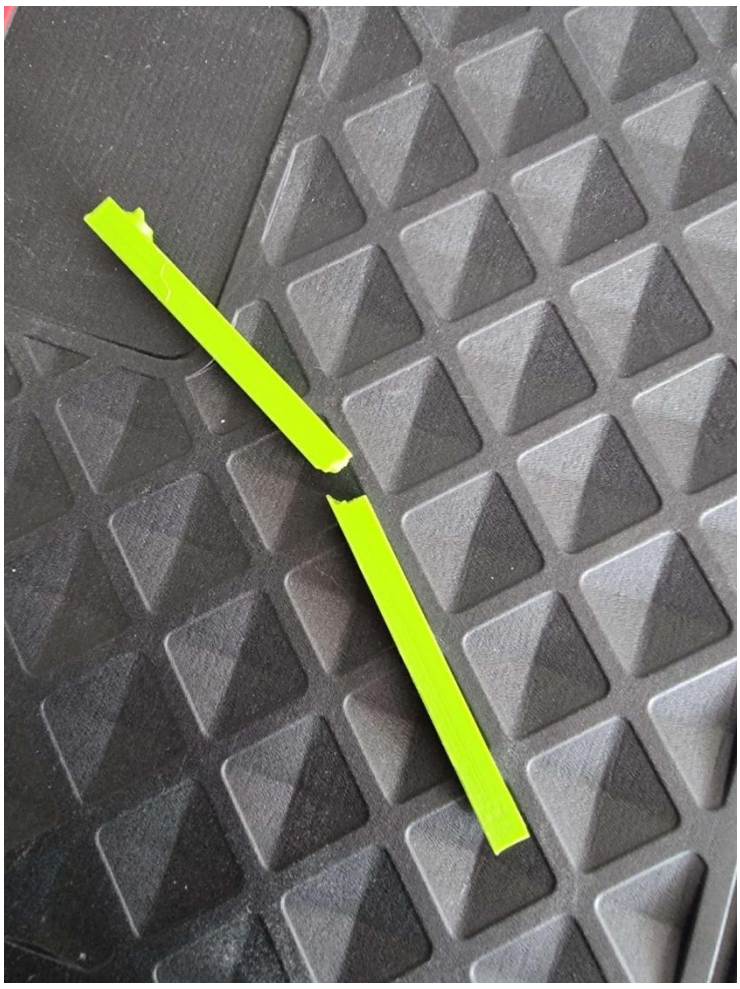


Figure 29 - Type I bar break

Some of the other testing came from having either old versions or spare parts that could be tested without detriment to development. This testing was helpful as it allowed elements that had remained unchanged between iterations, such as the main frame of the bicep brace, to be tested. This allowed us to test that element of the design while developing the other sections, and allowed for simultaneous design, fabrication and evaluation when appropriate.
During a partial systems test, we found that under a 10kg load, although there were cracking noises, it was suitable for use and inclusion in the design unchanged, as it exceeded the expected loading by the group for the prototype. During a systems check, it was also observed, as previously mentioned in the iterations section, that the left servo wing failed after a slight modification to one of the angles, yet the main body of the brace itself remained strong.

Following previous stages, test two (M-T02) yielded positive results for most components, indicating that friction did not pose a significant issue throughout the system. However, one exception was the main wrist collar, which required a slight adjustment. Due to the inherent imperfections in 3d-printed circular geometry, some play was necessary to ensure functional, reliable movement despite exporting STL files with high-resolution settings optimised for round features.

While lubrication was considered as a solution, allowing for minor clearance between mating parts ultimately proved more effective. This tolerance gave the wrist collar just enough freedom to rotate smoothly between the upper and lower bracket mounts without compromising positional stability. The part remained reliably seated during operation while maintaining its ability to move freely.

Material tests three and four (M-T03, M-T04) also passed successfully. During live demonstration, testing, and filming, the exoskeleton arm was worn and actuated with functional actuation for both elbow flexion and wrist movement. The system stayed securely in place on the user throughout the activity, confirming the validity of mounting elements under real-world conditions with the Velcro straps.

Electronic

This section outlines the electronic components and control logic used to drive the exoskeleton. It details how microcontrollers, sensors and power system were integrated to enable reliable, safe and responsive actuation

Approach

The electronics subsystem was designed to be modular, robust, and adaptable to ongoing design changes. It began with identifying the critical components required for safe and responsive actuation—servos, sensors, controllers, and power regulation—and arranging them in a way that allowed for iterative development and straightforward debugging.

At the heart of the system are three Arduino Uno boards, each handling a specific role: motor control via PWM, current monitoring with the INA266 sensor, and digital input processing from limit switches and emergency stop mechanisms. This decentralised approach offloaded low-level tasks from the Jetson Nano, which instead manages higher-level logic and wireless communication via MQTT.

Power was a central concern from the outset. To ensure stable operation, two 12V 6A batteries were used in parallel, feeding into a regulated power distribution setup. This configuration not only supported the servos' high torque demands but also allowed sufficient headroom for current spikes during rapid motion. Each servo line is protected against overcurrent events, with fail safe triggered both in software and through inline hardware mechanisms.

Wiring was kept as clean and maintainable as possible. Dupont jumper wires were avoided for high-load connections, replaced with soldered joints and secure terminal blocks where necessary. Serial communication between the Jetson Nano and each Arduino allowed reliable data exchange, while USB connections simplified both power delivery and debugging.

A primary focus was on ensuring that one subsystem's faults would not propagate to others. This was achieved by isolating power rails, using individual fuses, and monitoring system health continuously through telemetry data received from the INA266 and limit switch triggers.

The electronic layout evolved throughout the project as reliability issues emerged—particularly regarding I2C instability and early power delivery failures. Lessons from these

setbacks led to a more fault-tolerant design, with redundant safety layers and extensive logging via Node-RED and MQTT to allow for quick issue identification and resolution.

Every component—from relays and limit switches to servo headers—was tested in isolation before being integrated into the system. This strategy ensured that when errors did arise during full-system tests, they could be traced with minimal debugging overhead.

Iterations and Problems

The electronics subsystem underwent several iterations as the system evolved, and reliability issues surfaced during testing. Early designs relied heavily on the Jetson Nano for both processing and I2C communication, but persistent instability—particularly with I2C buses and peripheral interference—led to a shift toward decentralised control using three Arduino Unos. This change improved responsiveness and made debugging individual subsystems significantly easier.

One of the first challenges was voltage drop across long power lines, which caused intermittent servo resets under high load. This was mitigated by switching to thicker gauge wires and relocating power distribution closer to the load points. Similarly, early versions used off-the-shelf jumper wires, which often failed under vibration or torque; these were later replaced with soldered joints and screw terminals for reliability.

Power delivery itself posed a number of issues. In the initial prototype, all components shared a single rail without adequate current sensing or isolation. This led to system-wide brownouts when servos stalled. The introduction of individual INA266 current sensors and inline fusing helped isolate faults and prevent cascading failures. Combined with software logging over MQTT, these changes allowed for quicker identification of unsafe operating conditions.

The USB serial connection between the Jetson Nano and Arduinos also proved problematic in early tests, particularly when multiple devices attempted simultaneous communication. A structured messaging protocol and staggered polling intervals were introduced to prevent data collisions and ensure that sensor readings remained consistent across test cycles.

These iterative changes were driven by real-world testing and often revealed edge cases not caught during isolated component testing. While each revision introduced its own set of new issues, the end result was a robust control system that could be debugged quickly, modified easily, and scaled if necessary.

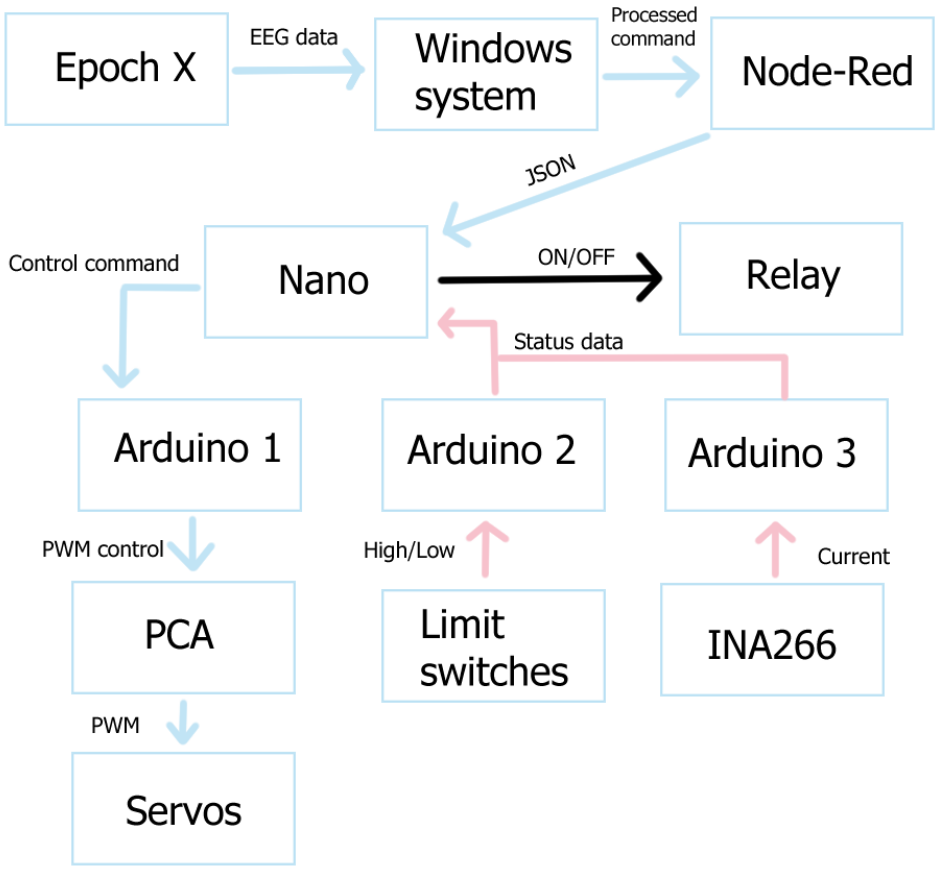


Figure 30 - Data Flow

This system diagram illustrates the full communication architecture of the exoskeleton control pipeline. EEG data is captured by the **Emotiv Epoch X** headset and sent to a **Windows system**, where it is processed and translated into action commands via **Node-RED**. These commands are formatted as JSON and transmitted to the **Jetson Nano**, which serves as the central control unit. The Nano distributes control instructions to three dedicated **Arduino boards**:

- **Arduino 1** receives servo angle commands and controls the **PCA9685**, which in turn drives the **servo motors**.
- **Arduino 2** monitors the state of **limit switches** and sends digital HIGH/LOW signals to indicate whether physical movement boundaries have been reached.
- **Arduino 3** reads real-time current values from two **INA266 sensors** to monitor power draw.

All status data from Arduinos 2 and 3 are sent back to the Jetson Nano, which uses this information to make safety decisions and can engage or disengage a physical **relay** that powers the system. The result is a modular, fault-tolerant architecture that allows real-time feedback, EEG control, and safe actuation of the exoskeleton.

Schematics and Circuit Diagrams

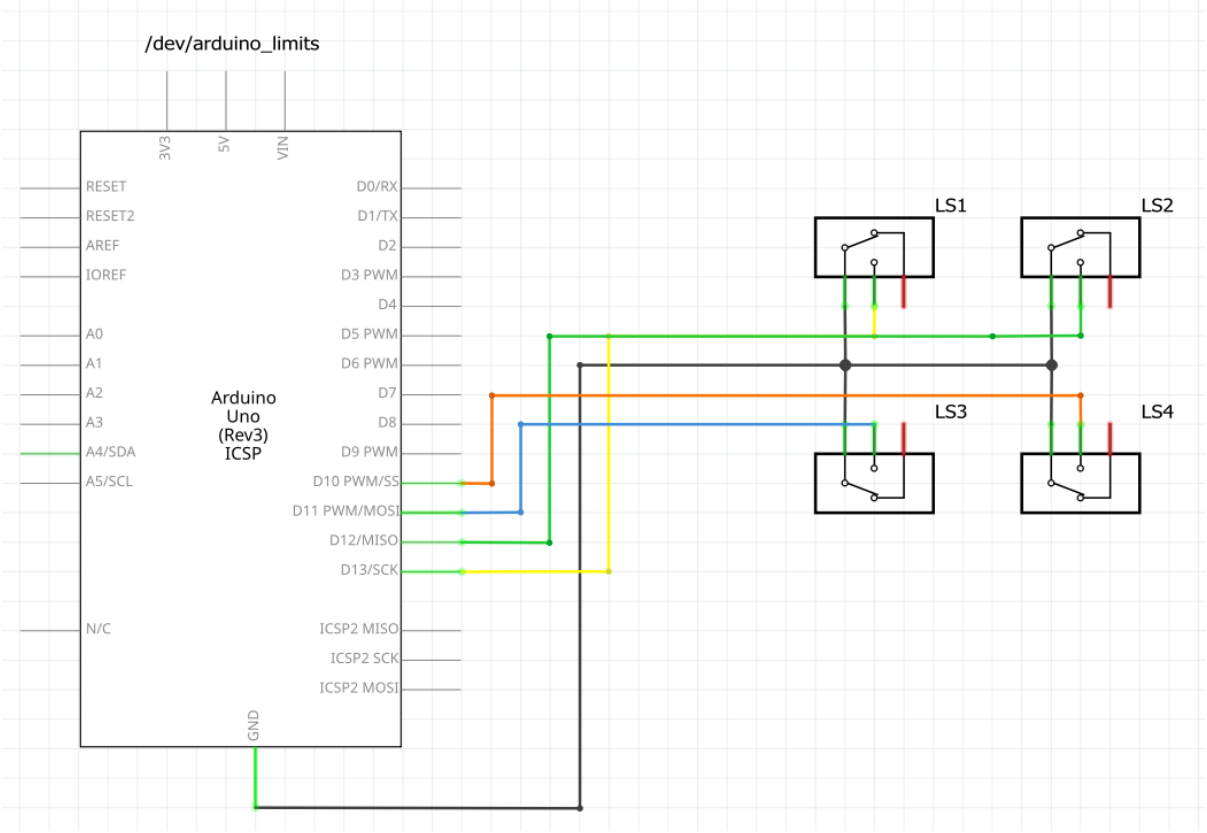


Figure 31 - Limit Switch Arduino Wiring Diagram

This diagram shows the wiring configuration for four mechanical limit switches (LS1–LS4) connected to an Arduino Uno designated as /dev/arduino_limits. The switches are used to detect end-of-range conditions in the exoskeleton's joints and enhance user safety. LS1 and LS2 are wired in parallel to digital pin D4, acting as redundant triggers for one axis (e.g. elbow flexion), while LS3 and LS4 are connected to pins D5 and D6 respectively, monitoring a second axis (e.g. wrist rotation). All switches share a common ground, ensuring consistent reference voltage. In the software, the Arduino uses internal pull-up resistors and monitors for pin state changes—when a switch is pressed, the pin reads LOW. This event is sent over serial to the Jetson Nano, where it can be interpreted by safety scripts to stop or reverse motor movement immediately. This setup provides a simple, reliable mechanism to enforce mechanical limits without the need for analog sensing or external logic.

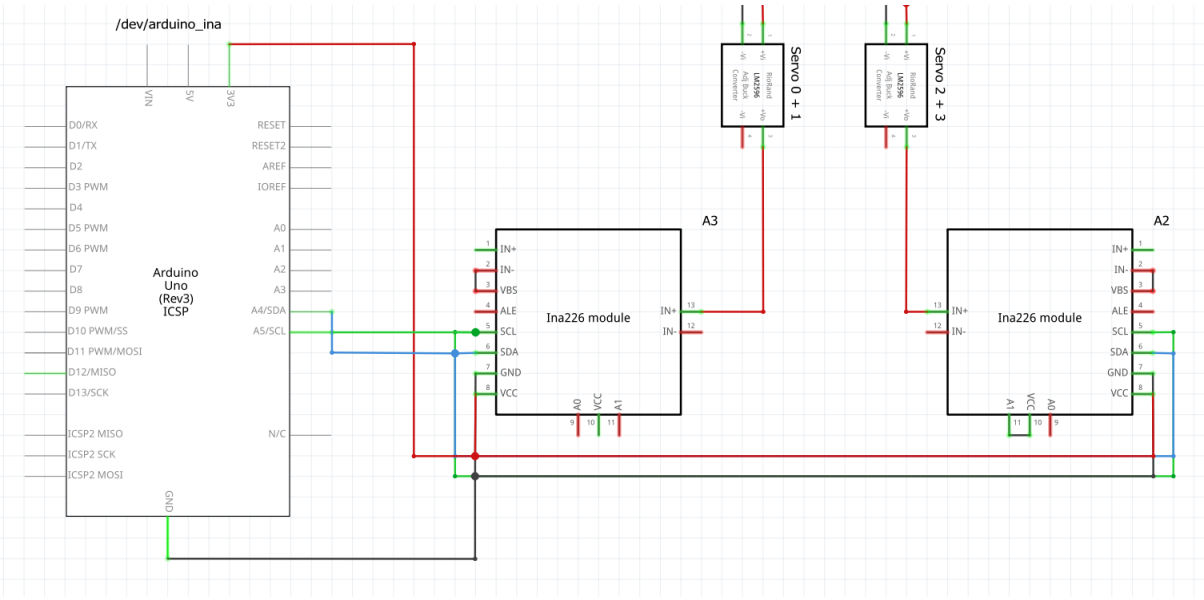


Figure 32 - INA Arduino Wiring Diagram

This diagram illustrates the current sensing setup using two **INA226 modules** connected to an Arduino Uno, referenced in software as /dev/arduino_ina. Each INA226 module monitors the current drawn by a pair of servo motors—**Servo 0 & 1** and **Servo 2 & 3**—providing real-time current feedback to enhance safety and system diagnostics. The modules communicate with the Arduino via the **I²C bus**, using shared SDA and SCL lines connected to **A4 and A5** on the Arduino. Each INA226 module is powered through the 5V and GND rails, and is placed in-line with the servo power supply to measure voltage and current directly across the shunt resistor. The analog inputs (**IN+ / IN-**) are connected in series with the servo power lines to enable precise sensing of load conditions. In software, the Arduino continuously reads current values from each module and transmits this data to the Jetson Nano via serial. These readings are used to detect overcurrent events and support real-time monitoring, allowing the control system to trigger power cut-offs or emergency stop routines if abnormal current draw is detected.

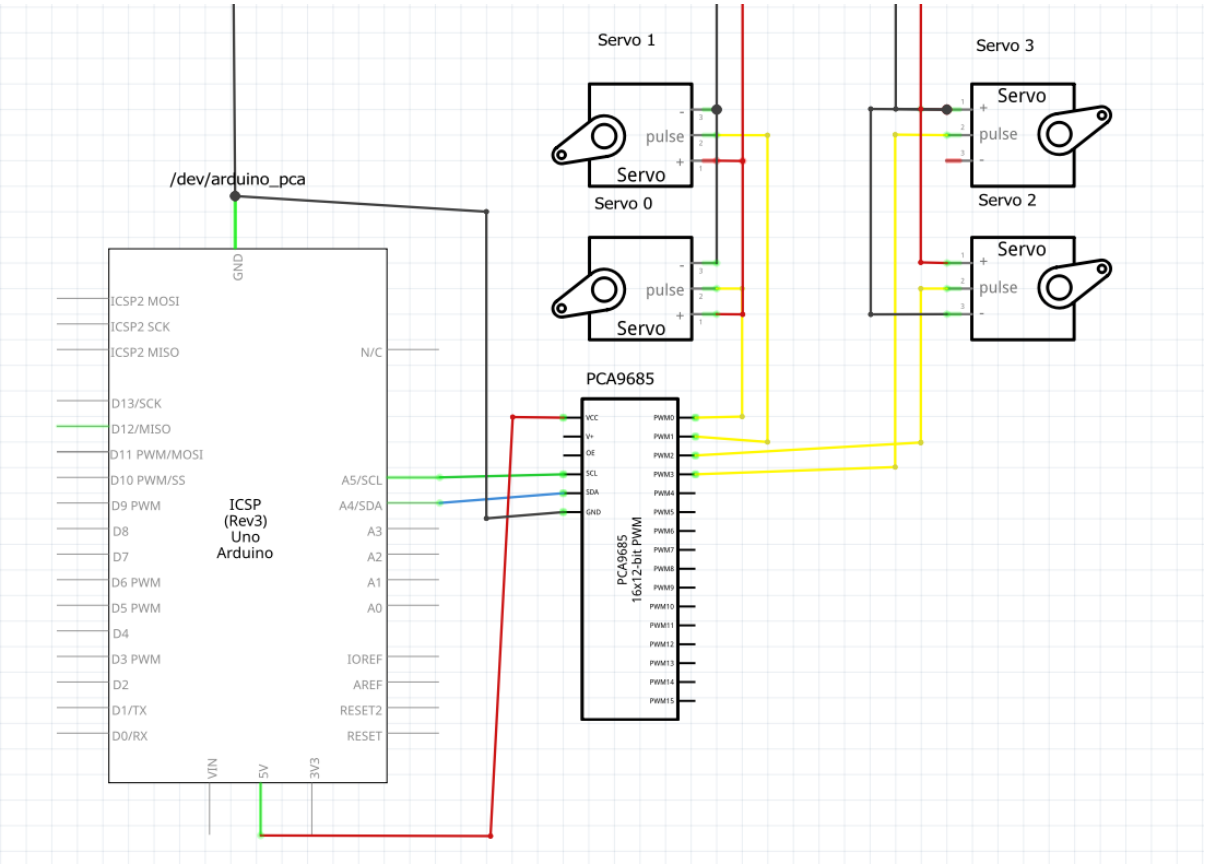


Figure 33 - PCA Arduino Wiring Diagram

This diagram shows the servo control setup using a **PCA9685 16-channel PWM driver** connected to an **Arduino Uno**, referenced in the system as /dev/arduino_pca. The PCA9685 receives control signals via **I²C communication**, with **SDA (A4)** and **SCL (A5)** connected to the Arduino. It is powered from the 5V rail and grounded alongside the Arduino to maintain a common reference. The PWM outputs from the PCA9685 are routed to four servos (Servo 0–3), with their **signal wires (yellow)** connected to channels **PWM0–PWM3** respectively.

Each servo receives **PWM control** from the PCA9685 while sharing a common **power (red)** and **ground (black)** line. The PCA9685 offloads the timing-intensive task of generating PWM signals, enabling smooth and simultaneous actuation of multiple servos without overloading the Arduino's internal timers. This configuration supports real-time motion control, with the Arduino forwarding angle commands from the Jetson Nano to the PCA9685 over I²C. To prevent missed signals during startup, a handshake mechanism is implemented: the Arduino sends an "ACK" message after its setup routine is complete, ensuring that the Jetson only begins communication once the system is ready. This design improves stability, especially in complex boot sequences and multi-script deployments.

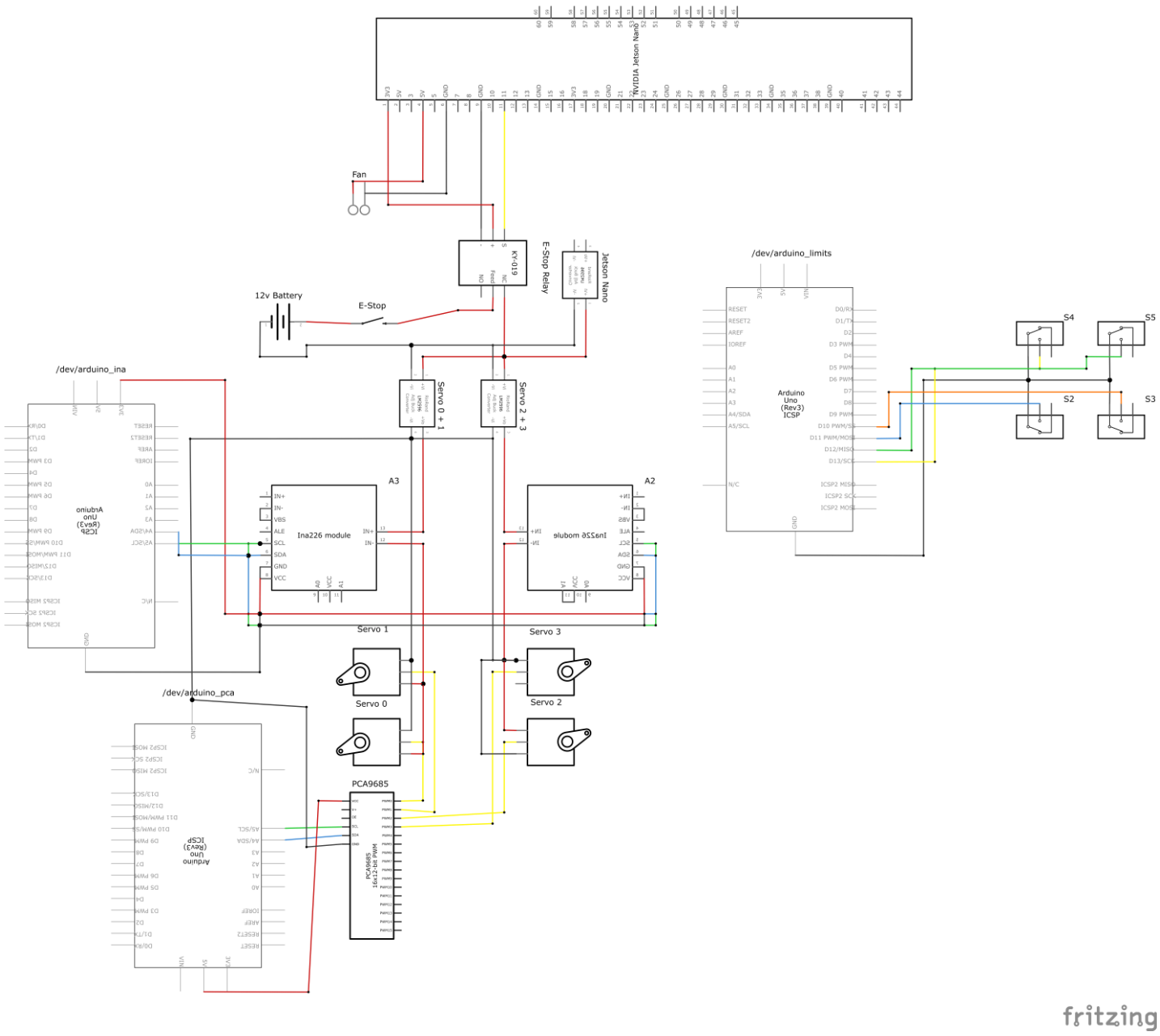


Figure 34 - Full Wiring Diagram

Justification of Final Design

The final electronic layout was driven by the need for modularity, fault isolation, and reliable performance under real-world conditions. While the original plan aimed for a more centralised system using the Jetson Nano for all control and processing, repeated communication errors, power inconsistencies, and debugging challenges made that approach unfeasible. Delegating low-level tasks to dedicated Arduino Unos allowed for cleaner separation of responsibilities, reduced wiring complexity, and simplified troubleshooting.

This decentralised architecture was particularly valuable in ensuring each subsystem—PWM control, current monitoring, and limit switch logic—could operate independently and be tested in isolation. The USB serial links provided a reliable communication method with the Jetson Nano, which now handled high-level decision-making and external MQTT messaging. This setup proved effective in both bench tests and live demonstrations, with individual faults no longer causing total system failure.

Using dual 12V 6A batteries ensured sufficient current headroom for the high-torque servos and eliminated brownouts observed during earlier testing phases. The addition of INA266 sensors allowed for real-time current monitoring, improving safety and offering critical feedback during load testing. Safety was further reinforced with physical relays for emergency stop functionality, inline fuses, and software-triggered shutdowns that could cut power instantly in the event of an overcurrent or system hang.

Soldered joints replaced all high-load or vibration-sensitive connections to minimise signal loss and physical disconnection. All PCB-less wiring was routed with strain relief in mind, using screw terminals, heat shrink, and zip ties to maintain order and reduce wear during extended testing periods.

Overall, the final design balances reliability, modularity, and real-world usability. While more compact or integrated systems are feasible with advanced PCBs or embedded solutions, the chosen approach prioritised accessibility, rapid iteration, and maintainability—key factors in a team-led, proof-of-concept build.

Software

This section details the software architecture and tools used to control the exoskeleton, interpret EEG signals, and manage communication between system components. The software was developed with modularity, fault tolerance and ease of debugging, using Python scripts, Node-RED flows and MQTT messaging.

Approach

The software was designed to bridge multiple hardware layers flexibly and resiliently. Rather than relying on a single monolithic application, the system was broken into smaller, purpose-built Python scripts—each handling a dedicated function such as servo control, EEG command interpretation, or current monitoring. This modular approach allowed for faster debugging, easier updates, and improved fault isolation across the project's lifespan.

At the system's core was a lightweight communication framework built on MQTT. MQTT topics were routed messages between the EEG input, control scripts, and safety mechanisms. The broker was hosted in a Docker container on a Windows laptop, ensuring compatibility and isolation from the host environment. Node-RED served as both a flow-based development tool and a user interface, providing a clear visual overview of signal routing and system state.

EEG signal data was processed using Emotiv's cloud services, which provided structured JSON outputs in response to trained mental commands. These outputs were subscribed to via Node-RED and relayed through MQTT to the Jetson Nano, triggering corresponding motor commands. By using this pipeline, the team avoided the need for local signal processing—reducing computational load on the Jetson Nano and simplifying the development stack.

Due to instability and delays in setting up ROS and Gazebo, a decision was made early in development to drop the ROS-based control layer entirely. Instead, the system adopted a lightweight custom alternative using MQTT and Python, which allowed faster iteration and fewer compatibility issues. This trade-off proved effective for a prototype-level project, where flexibility and reliability were more important than long-term scalability or ROS-native features.

Each script was developed to run independently and report its status through MQTT topics, making it easier to isolate faults or run subsystems in simulation if needed. For example, the servo controller could be tested with mock EEG messages, or the current sensor could be monitored in real time using a separate logging script. This structure also allowed easy extension of features without rewriting the whole system.

The software stack included minimal dependencies to maintain portability. Python was selected as the primary language due to its extensive library support, readability, and compatibility with both Jetson Nano and Windows environments. Supporting tools such as Docker, Mosquitto, Node-RED, and PowerShell scripts were all chosen to streamline system deployment, testing, and monitoring.

Ultimately, the software was designed not just to function but also to be understandable, adaptable, and recoverable. These priorities shaped the system's structure and helped ensure it remained operable throughout iterative hardware changes and evolving requirements.

Iterations and Problems

A major challenge in the software development process was the unpredictable reassignment of serial ports for the Arduinos connected to the Jetson Nano. Each time the system rebooted or an Arduino was unplugged and reconnected, Linux dynamically assigned it a different device path (e.g. /dev/ttyACM0, /dev/ttyACM1, etc.). This caused significant issues for the modular Python scripts, which required consistent communication with specific Arduinos responsible for motor control, current sensing, and limit switch handling.

Initially, port assignments had to be manually updated in each script before launch. However, this quickly became error-prone and time-consuming—especially as incorrect assignments often led to silent failures, such as the wrong Arduino being sent PWM commands or sensor data being misread. To resolve this, we implemented a permanent fix using udev rules. By identifying each Arduino's unique serial number using udevadm info, we created persistent symbolic links such as /dev/arduino_pca, /dev/arduino_current, and /dev/arduino_limits. These symlinks acted as reliable placeholders in all scripts, ensuring consistent behaviour regardless of device order or boot timing.

Another critical issue emerged from the Arduino handling the PCA9685 board. Due to slight startup delays and inconsistent boot times, the main Python script would sometimes begin transmitting commands before the Arduino was fully ready to receive them. This resulted in dropped messages and delayed servo actuation during system startup. To mitigate this, we implemented a custom handshake protocol between the main script and the PCA Arduino.

At startup, the Python script waits for a specific "ACK" message from the Arduino over serial before sending any PWM commands. This ensures the Arduino has completed its setup routine and is actively listening. Conversely, the Arduino remains in a passive state until it sends the acknowledgment, preventing it from missing the first command. This small addition proved highly effective in improving reliability, especially during repeated system tests where power cycling was frequent.

Other software issues included occasional serial buffer overruns and inconsistent data parsing when too many messages were sent in rapid succession. These were addressed by rate-limiting serial writes, flushing buffers before read cycles, and introducing short delays between message transmissions. The modular structure of the software also made debugging more manageable, as individual scripts could be run and monitored in isolation to pinpoint faults.

Overall, these iterations significantly improved system robustness. What began as a loosely connected set of scripts matured into a coordinated, resilient control system with clear communication pathways, startup synchronisation, and fault tolerance—all essential for real-world operation of the exoskeleton.

Justification of Final Design

The final software architecture was intentionally built around simplicity, modularity, and fault tolerance—key traits for a prototype system where hardware and requirements evolved frequently. Rather than relying on a monolithic framework like ROS, which introduced delays and compatibility issues, the team opted for lightweight Python scripts communicating over MQTT. This choice allowed faster development, easier debugging, and greater flexibility in integrating or isolating subsystems during testing.

By structuring each software component as an independent script—handling servo control, EEG interpretation, or sensor monitoring—errors in one area did not compromise the entire system. This was essential for iterative development and real-time testing, where subsystems could be swapped or restarted without halting the overall workflow. Node-RED provided an accessible interface for monitoring system state and routing messages, making the control logic more transparent and adaptable.

The decision to use Emotiv's cloud-based EEG processing simplified signal handling and reduced the computational load on the Jetson Nano. Instead of building a custom signal processing pipeline locally, the team focused on interpreting already-processed commands, which allowed greater emphasis on system integration and reliability. This trade-off also reduced potential technical debt and avoided the complexity of working with proprietary EEG data formats.

Persistent symlinks for serial communication were another critical design decision. They ensured that each Arduino could be referenced reliably by name—such as /dev/arduino_pca—regardless of USB enumeration order. This was especially important for scripts running on boot or in timed sequences, where incorrect device assignment could cause unpredictable behaviour. Combined with a startup handshake protocol, these symlinks helped establish a more deterministic and repeatable system state at runtime.

Finally, the lightweight, decoupled nature of the software stack aligned with the project's overall goals: to demonstrate EEG-based control of a wearable exoskeleton using accessible, open tools. While not optimised for scalability or embedded efficiency, the final design was stable, understandable, and maintainable meeting its core requirements for real-world testing and demonstration.

Node-RED

When setting up the Node-RED flows for EEG-based control, a couple of node palettes had to be installed to enable functionality. Among these, the Emotiv-BCI nodes were critical for interfacing with the Emotiv headset, while the dashboard 1.0 palette allowed for real-time monitoring and visualisation of system states and command outputs. These tools were fundamental to creating a control and monitoring interface; allowing observing of the system's users' control signals without needing to rely on watching Emotiv Suite directly all the time.

The flow design underwent numerous iterations throughout the development cycle. Initially, the objective was simply to establish communication with the headset and begin receiving data within the Node-RED environment. This early stage was exploratory, helping us understand the structure and characteristics of the data being streamed from Emotiv's Cortex API, and what processing would be necessary before transmitting control signals to the exoskeleton over MQTT.

One of the first improvements was the inclusion of function nodes to filter and format incoming data. Emotiv's API, in its free tier, limits user access to only trained mental commands - delivered as intensity values on a 0 to 100 scale. These commands had to be thresholded to prevent false positives and misfires. A script was implemented to parse incoming payloads and apply conditional logic: if the intensity exceeded a predefined

threshold, a corresponding command such as "lift" was passed forward; otherwise, a "none" command was issued.

This filtering was essential not only for accuracy but also to mitigate a soon to be known Node-RED quirk: even if a script doesn't explicitly return an output, unhandled data can still propagate through the flow. Without returning "none", stray messages - in the form of zeroes - were transmitted, leading to possible confusion for an unexpected input and no way to handle it cleanly. Including the explicit "none" command ensured consistent, intentional communication over MQTT as we had a known message to ignore.

After processing, the cleaned command data was transmitted over MQTT, where the Jetson Nano subscribed to the appropriate topics and used these commands to actuate servos in the exoskeleton.

Working with Emotiv's Cortex API introduced several technical challenges. A particularly persistent issue was that the Emotiv suite software had to be launched before starting Node-RED - failing to do so would cause WebSocket connection errors, with the API being marked as "busy" or not attempting to connect at all. Additionally, a later update to Emotiv's license agreement broke compatibility with containerised environments, meaning that our Node-RED instance, which had previously been deployed via Docker, now failed to connect. The workaround was to run Node-RED natively on the host machine, rather than within a container, while continuing to run Mosquitto in Docker.

Once the connectivity issues were resolved, final improvements could be made to the Node-RED dashboard. These included enhancements to usability and user feedback, such as displaying the last sent command and its intensity in textual form and switching from static gauges to real-time graphs. This provided a more informative overview of cognitive activity, allowing the user or an observer to see fluctuations in thought strength and control quality without having to monitor the Emotiv software directly.

The dashboard was also designed with safety and user control in mind. A pair of buttons were added to control a software-based emergency stop (E-Stop). These acted as toggles, emitting Boolean values on press. The visual design featured a large red button to engage the E-Stop and a green one to disengage it. Between these, a power indicator was placed, dynamically changing from green (active) to red (stopped) to clearly show the system's state at a glance—eliminating the need for the user to remember which button was pressed last.

Realisation of Evaluation of the Final Robotic System

This section presents the final integrated prototype and assesses its meeting the project's original objectives. Based on real-world testing, it reflects on system performance, reliability, and usability, highlighting both the developed exoskeleton's successes and limitations.

Software Realisation

The software realisation focused on building a modular, lightweight, and fault-tolerant control system that reliably translates EEG commands into physical movement. The architecture was developed around a set of dedicated Python scripts that each handled specific tasks, such as receiving mental command signals, controlling servo positions, monitoring system current, and managing safety responses. These scripts communicated via MQTT topics, allowing each subsystem to operate independently while remaining synchronised through a shared messaging structure.

Key to the software's success was the decision to avoid using a full ROS-based stack. ROS introduced unnecessary complexity and compatibility issues during early testing, especially when simulation efforts with Gazebo stalled. Instead, MQTT and Node-RED provided a more accessible and stable alternative, allowing visual monitoring of signal flow and real-time debugging during development.

Each Arduino was addressed through persistent symlinks based on serial numbers, avoiding issues caused by dynamic USB enumeration. A handshake protocol was also added to ensure that the PCA9685 Arduino was fully initialised before accepting servo commands, addressing lag and missed messages during startup.

The final software structure prioritised clarity and reliability. Components could be swapped or updated without affecting the overall system, supporting ongoing hardware changes and simplifying maintenance. This approach enabled smooth testing sessions, quick recovery from faults, and a clear demonstration of EEG-based actuation for rehabilitation use.

Software Development

The software development process was structured to support remote collaboration, rapid iteration, and seamless deployment across heterogeneous hardware. Development took place primarily on a Windows laptop, with remote access to the Jetson Nano—running the main control software—established via SSH. This enabled the team to manage code, monitor processes, and push updates without requiring direct interaction with the Jetson's desktop interface, which remained headless throughout the project.

The team used Visual Studio Code in combination with the Remote - SSH extension, which allowed the Jetson Nano's file system to be accessed as if it were local. This setup provided full IDE functionality—including syntax highlighting, Git integration, and terminal access—while keeping execution and testing bound to the Jetson. It was especially helpful when debugging live systems, as logs could be streamed in real time while viewing or editing the code directly responsible.

Python was chosen as the primary development language for its simplicity, wide hardware support, and the availability of essential libraries such as paho-mqtt, pyserial, and json. Each component of the system—servo control, EEG command handling, safety logic, and current sensing—was developed as a standalone script. This modular approach allowed subsystems to be launched, stopped, and debugged independently, reducing interdependency and making it easier to isolate faults during testing.

To manage communication between these components, the team implemented MQTT using the Mosquitto broker. The broker was hosted inside a Docker container on the Windows laptop, ensuring environment consistency and allowing the system to be brought up or restarted reliably. Other containerised services included the EEG interface (via Node-RED) and logging dashboards. This decision was made to avoid the complexity of cross-platform dependency issues, as containerising the services enabled reproducible environments across team machines.

Node-RED was used not just for EEG signal routing, but also as a high-level visual debugging tool. It provided real-time status indicators and flow control, making it easier to track whether data was reaching its intended destination. In practice, this meant commands from the Emotiv system could be monitored in a clear, web-based UI before being passed to the Python control scripts.

As development progressed, the team encountered recurring issues with Arduino device enumeration—specifically, that each board (connected via USB) would randomly be assigned paths such as /dev/ttyACM0, /dev/ttyACM1, etc. This inconsistency made automated scripts prone to failure, as the mapping between logical role and device path could shift between boots. To address this, we used udev rules to assign persistent symlinks to each Arduino based on its unique serial number. These were given intuitive names like /dev/arduino_pca, /dev/arduino_current, and /dev/arduino_limits, which were then referenced in the software. This guaranteed reliable connections to the correct microcontroller, regardless of connection order or timing.

In addition, a serial handshake protocol was implemented specifically for the Arduino responsible for driving the PCA9685 servo controller. During initial tests, the Jetson script would begin sending commands before the Arduino had finished initialising, causing delays or missed instructions. To resolve this, the Arduino was programmed to send an "ACK" message once it was ready, and the Jetson's Python script would wait for this signal before transmitting any commands. This simple but effective protocol ensured synchronisation between startup routines and improved overall system responsiveness.

Version control was managed using Git, with local repositories kept on both the Jetson and the development laptop. Changes were synced via SSH, allowing branches to be tested safely before merging. This helped track changes over time and provided a recovery mechanism in the event of breaking updates.

Altogether, the software development strategy focused on minimising friction between writing code and seeing it operate in real-world conditions. By leaning on remote tooling, containerisation, and modular design, the team maintained development velocity even as the hardware configuration and project scope evolved. This setup was instrumental in delivering

a working prototype that could reliably interpret EEG signals and perform real-time actuation under live test conditions.

Important Code Extracts

To ensure consistent serial communication with each Arduino, persistent device naming was implemented using udev rules. This process begins by plugging in the Arduino via USB and identifying its unique serial number using the command udevadm info -a -n /dev/<device> | grep '{serial}' | head -n 1. Once the serial number is obtained, a custom rules file (99-arduino-names.rules) is created in /etc/udev/rules.d/, where each Arduino is assigned a symbolic link name (e.g. arduino_pca, arduino_ina, arduino_limits) based on its serial ID. These rules ensure that, regardless of the order in which Arduinos are connected or rebooted, each one is assigned a predictable path in /dev/. After saving the rules, they are reloaded and triggered using udevadm, and verification is done by listing the resulting symlinks with ls -l /dev/arduino_*. This setup eliminates ambiguity in device assignment and allows the main control scripts to reliably communicate with the correct microcontroller every time the system starts.

To automate system startup and ensure all critical scripts launch on boot, a custom desktop autostart entry and a Bash script were used. A .desktop file was created in /home/<USER>/.config/autostart, which references a script (startup.sh) that sequentially opens terminal windows and executes each required Python module. This includes the scripts for controlling the PCA9685 (mqtt_PCA9685.py), current sensing (mqtt_INA226.py), limit switch monitoring (limits_serial_mqtt.py), the main logic controller (main.py), and the software emergency stop handler (soft_estop.py). Each command is launched in a new GNOME terminal instance with a small delay between them to avoid resource contention during initialisation. This method provides a simple, reliable way to automatically bring the entire exoskeleton control system online each time the Jetson Nano starts, supporting consistent operation without requiring manual intervention.

```
# startup.sh
#!/bin/bash

# Navigate to the directory where your Python scripts are
cd /home/jetson/exoskeleton

# Launch each script in a new terminal window in order
gnome-terminal -- bash -c "python3 mqtt_PCA9685.py; exec bash"
sleep 1
gnome-terminal -- bash -c "python3 mqtt_INA226.py; exec bash"
sleep 1
gnome-terminal -- bash -c "python3 limits_serial_mqtt.py; exec bash"
sleep 1
gnome-terminal -- bash -c "python3 main.py; exec bash"
sleep 1
gnome-terminal -- bash -c "python3 soft_estop.py; exec bash"
```

[Desktop Entry]

Type=Application

Exec=/home/jetson/exoskeleton/startup.sh

Hidden=false

NoDisplay=false

X-GNOME-Autostart-enabled=true

Name=Start Exoskeleton

Comment=Starts all Python scripts at login, this file needs to be saved in /home/<USER>/.config/autostart

INA226 Current Sensing

This Arduino sketch interfaces with two INA226 current sensors over I²C to monitor the voltage and current drawn by the system's servo power rails. Each sensor is calibrated for a 0.1 Ω shunt resistor with a maximum expected current of 4.5 A, ensuring accurate readings without automatic range normalisation. In the main loop, the Arduino reads both voltage and current values from each INA226 module and outputs them over serial in structured JSON format. This data is read by the Jetson Nano via a Python script, then published over MQTT for real-time monitoring, overcurrent detection, and logging. The update rate is set to once per second, balancing responsiveness with communication overhead.

```
#include <Wire.h>
#include <INA226.h>

INA226 ina1(0x40);
INA226 ina2(0x41);

void setup() {
  Serial.begin(115200);
  Wire.begin();

  ina1.begin();
  ina2.begin();

  // Calibrate both sensors for 0.1Ω shunt and 4.5A max expected current
  // normalise = false ensures full range (no rounding)
  int err1 = ina1.setMaxCurrentShunt(4.5, 0.1, false);
  int err2 = ina2.setMaxCurrentShunt(4.5, 0.1, false);

  if (err1 != INA226_ERR_NONE) {
    Serial.print("INA1 Calibration Error: 0x");
    Serial.println(err1, HEX);
  }
  if (err2 != INA226_ERR_NONE) {
    Serial.print("INA2 Calibration Error: 0x");
```

```
    Serial.println(err2, HEX);
  }
}

void loop() {
  float v1 = ina1.getBusVoltage();      // V
  float i1 = ina1.getCurrent();         // A

  float v2 = ina2.getBusVoltage();      // V
  float i2 = ina2.getCurrent();         // A

  Serial.print("{\"ina1\": {\"voltage\": ");
  Serial.print(v1, 3);
  Serial.print(", \"current\": ");
  Serial.print(i1, 3);
  Serial.print("}, \"ina2\": {\"voltage\": ");
  Serial.print(v2, 3);
  Serial.print(", \"current\": ");
  Serial.print(i2, 3);
  Serial.println("}}");

  delay(1000);
}
```

Limit Switch Monitoring

This sketch reads the state of four limit switches wired to digital pins 8–11 and sends their stable states over serial in JSON format every 50ms. Internal pull-up resistors are used, so each switch is active LOW when pressed. A simple software debounce mechanism filters out signal noise by requiring a 30ms stable reading before considering the switch state changed. The output JSON includes boolean values for LS1 through LS4, allowing the Jetson Nano to monitor joint boundaries and respond appropriately—such as stopping or reversing servo motion if a physical limit is reached. This approach ensures safety through continuous, low-latency monitoring.

```
const int switchPins[4] = {8, 9, 10, 11};
bool currentStates[4] = {false, false, false, false};
bool stableStates[4] = {false, false, false, false};
unsigned long lastReadTime[4] = {0, 0, 0, 0};
const int debounceDelay = 30;  // Minimum stable time in ms

void setup() {
  Serial.begin(115200);
  for (int i = 0; i < 4; i++) {
    pinMode(switchPins[i], INPUT_PULLUP); // Using internal pull-up resistors
  }
}
```

```
void loop() {
  unsigned long now = millis();

  for (int i = 0; i < 4; i++) {
    bool reading = digitalRead(switchPins[i]) == LOW;  // true if pressed

    if (reading != currentStates[i]) {
      // State changed - start debounce timer
      lastReadTime[i] = now;
      currentStates[i] = reading;
    }

    if ((now - lastReadTime[i]) > debounceDelay) {
      // State stable - update
      stableStates[i] = currentStates[i];
    }
  }

  // Send JSON output every 50ms
  static unsigned long lastSend = 0;
  if (now - lastSend > 50) {
    Serial.print("{");
    Serial.print("\"LS1\":"); Serial.print(stableStates[0] ? "true" :
"false"); Serial.print(",");
    Serial.print("\"LS2\":"); Serial.print(stableStates[1] ? "true" :
"false"); Serial.print(",");
    Serial.print("\"LS3\":"); Serial.print(stableStates[2] ? "true" :
"false"); Serial.print(",");
    Serial.print("\"LS4\":"); Serial.print(stableStates[3] ? "true" :
"false");
    Serial.println("}");
    lastSend = now;
  }
}
```

PCA9685 Servo Control

This sketch controls up to 16 servo channels using the PCA9685 PWM driver, communicating over I²C. The Arduino receives servo angle commands in JSON format via serial input (e.g., { "Servo0": 90 }), parses them using the ArduinoJson library, and maps the angles (0–180°) to corresponding PWM pulse widths (500–2500 µs). The servo pulses are then written to the appropriate channel on the PCA9685, enabling real-time control of multiple joints. The sketch includes feedback via the serial monitor to confirm each command, and serves as the core actuation interface in the exoskeleton's modular control system.

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
#include <ArduinoJson.h>

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x40);

// Servo pulse range
const int SERVO_MIN = 102;  // ~500us
const int SERVO_MAX = 512;  // ~2500us

void setup() {
  Serial.begin(115200);
  Wire.begin();
  pwm.begin();
  pwm.setPWMFreq(50);  // Standard servo frequency
  delay(10);
}

void loop() {
  static String input;
  while (Serial.available()) {
    char c = Serial.read();
    if (c == '\n') {
      processJson(input);
      input = "";
    } else {
      input += c;
    }
  }
}

void processJson(const String& jsonStr) {
  StaticJsonDocument<128> doc;
  DeserializationError err = deserializeJson(doc, jsonStr);
  if (err) {
    Serial.print("JSON parse error: ");
    Serial.println(err.c_str());
    return;
  }

  for (JsonPair kv : doc.as<JsonObject>()) {
    String key = kv.key().c_str();
    int value = kv.value().as<int>();

    if (key.startsWith("Servo")) {
      int channel = key.substring(5).toInt(); // "Servo1" → 1
      value = constrain(value, 0, 180);
      int pulse = map(value, 0, 180, SERVO_MIN, SERVO_MAX);
```

```
      pwm.setPWM(channel, 0, pulse);
      Serial.print("Set ");
      Serial.print(key);
      Serial.print(" to ");
      Serial.print(value);
      Serial.println(" degrees");
    }
  }
}
```

Python E-stop Relay Script

This Python script runs on the Jetson Nano and manages a physical emergency stop relay connected via GPIO. It listens for MQTT messages on the topic jetson/estop/relay, expecting a JSON payload with an estop boolean value. When estop is true, the script immediately disables the relay by setting the GPIO pin HIGH, cutting power to the system. When estop is false, a 3-second confirmation timer begins—ensuring transient errors or noise do not cause unintended system reactivation. After 3 seconds without interruption, the relay is re-enabled (GPIO LOW). The script also publishes a status heartbeat to the jetson/status/run topic on successful connection. This mechanism adds an essential safety layer, ensuring that emergency stop conditions are honoured with both hardware enforcement and time-based stability.

```
import Jetson.GPIO as GPIO
import time
import json
import paho.mqtt.client as mqtt
import threading

# === GPIO Setup ===
GPIO.setmode(GPIO.BOARD)
RELAY_PIN = 11  # GPIO17 (with internal pull-up on Jetson)
GPIO.setup(RELAY_PIN, GPIO.OUT)
GPIO.output(RELAY_PIN, GPIO.HIGH)  # Start with relay OFF (active LOW)

# === MQTT Setup ===
BROKER = "10.125.124.177"
PORT = 1883
TOPIC = "jetson/estop/relay"
STATUS_TOPIC = "jetson/status/run"

# === Relay Logic ===
class RelayController:
    def __init__(self):
        self.confirmed_estop = True  # Start in estop state
        self.timer = None
        self.lock = threading.Lock()
        GPIO.output(RELAY_PIN, GPIO.HIGH)  # Ensure relay is OFF initially
```

```
    def update_estop(self, new_estop):
        with self.lock:
            if new_estop:
                if self.timer:
                    self.timer.cancel()
                    self.timer = None
                if self.confirmed_estop is not True:
                    self.confirmed_estop = True
                    self._apply_relay_state(True)
            else:
                if self.timer:
                    self.timer.cancel()
                self.timer = threading.Timer(3.0, self._confirm_estop_clear)
                self.timer.start()

    def _confirm_estop_clear(self):
        with self.lock:
            if self.confirmed_estop != False:
                self.confirmed_estop = False
                self._apply_relay_state(False)

    def _apply_relay_state(self, estop):
        if estop:
            print("E-Stop engaged → Relay OFF")
            GPIO.output(RELAY_PIN, GPIO.HIGH)
        else:
            print("E-Stop cleared → Relay ON")
            GPIO.output(RELAY_PIN, GPIO.LOW)

    def cleanup(self):
        if self.timer:
            self.timer.cancel()
        GPIO.output(RELAY_PIN, GPIO.HIGH)
        GPIO.cleanup()

relay_controller = RelayController()

# === MQTT Callbacks ===
def on_connect(client, userdata, flags, rc):
    print("Connected to MQTT broker")
    client.subscribe(TOPIC)
    # Publish run status
    client.publish(STATUS_TOPIC, json.dumps({"run": True}))

def on_message(client, userdata, msg):
    try:
        payload = json.loads(msg.payload.decode())
```

```
        estop = payload.get("estop")
        if isinstance(estop, bool):
            relay_controller.update_estop(estop)
        else:
            print("Invalid payload:", payload)
    except Exception as e:
        print("Error parsing message:", e)

# === Main ===
client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message

try:
    client.connect(BROKER, PORT, 60)
    client.loop_forever()
except KeyboardInterrupt:
    print("\nShutting down...")
finally:
    relay_controller.cleanup()
```

Python Servo Control Script

This Python script bridges MQTT-based control messages to the Arduino responsible for servo actuation via the PCA9685 driver. It listens to the topic jetson/command/servo, expecting JSON payloads with keys like "servo1" and "servo2", which are internally remapped to PCA9685 channels ("Servo0", "Servo1"). Upon receiving a command, the script serialises the data into JSON and transmits it over /dev/arduino_pca using a dedicated lock to ensure thread-safe communication. It waits for an "OK" response from the Arduino before proceeding, providing a basic form of handshake confirmation. This script ensures real-time, reliable motor control in response to user intent, whether triggered by EEG commands or other high-level logic within the system.

```
import json
import time
import serial
import threading
import paho.mqtt.client as mqtt

# === MQTT Config ===
MQTT_BROKER = "10.125.124.177"
MQTT_PORT = 1883
MQTT_TOPIC = "jetson/command/servo"

# === Serial Config ===
SERIAL_PORT = "/dev/arduino_pca"
BAUDRATE = 115200
ser = serial.Serial(SERIAL_PORT, BAUDRATE, timeout=1)
```

```
time.sleep(2)  # Wait for Arduino reset

# === Lock for serial access ===
lock = threading.Lock()

def send_to_arduino(payload):
    try:
        with lock:
            ser.write((json.dumps(payload) + "\n").encode("utf-8"))
            while True:
                line = ser.readline().decode("utf-8").strip()
                if line == "OK":
                    break
    except Exception as e:
        print("Serial error:", e)

# === MQTT Callbacks ===
def on_connect(client, userdata, flags, rc):
    print("Connected to MQTT broker with result code", rc)
    client.subscribe(MQTT_TOPIC)

def on_message(client, userdata, msg):
    try:
        payload = json.loads(msg.payload.decode("utf-8"))
        mapped = {}
        for k, v in payload.items():
            if k == "servo1":
                mapped["Servo0"] = v
            elif k == "servo2":
                mapped["Servo1"] = v
        if mapped:
            send_to_arduino(mapped)
            print(f"Sent to Arduino: {mapped}")
    except Exception as e:
        print("Failed to handle message:", e)

# === Start MQTT client ===
client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message
client.connect(MQTT_BROKER, MQTT_PORT, 60)

try:
    client.loop_forever()
except KeyboardInterrupt:
    print("Exiting...")
    ser.close()
```

Python INA226 Current Monitoring Script

This script reads real-time voltage and current data from an Arduino connected to two INA226 sensors via I²C. The Arduino sends JSON-formatted data over serial to the Jetson Nano, where this script parses the output and publishes it to MQTT topics jetson/monitor/power/ina1 and jetson/monitor/power/ina2. Each loop cycle reads a single line of serial input, decodes the JSON payload, and forwards the respective INA1 and INA2 readings to the MQTT broker at 10.125.124.177. Error handling is included for serial communication issues and malformed JSON, ensuring stable long-term operation. This setup provides essential power telemetry for live monitoring, debugging, and safety validation of the exoskeleton's electrical system.

```
import serial
import json
import time
import paho.mqtt.client as mqtt

# Config
SERIAL_PORT = "/dev/arduino_ina"
BAUD_RATE = 115200
MQTT_BROKER = "10.125.124.177"
MQTT_PORT = 1883
TOPIC_INA1 = "jetson/monitor/power/ina1"
TOPIC_INA2 = "jetson/monitor/power/ina2"

# Setup MQTT
client = mqtt.Client()
client.connect(MQTT_BROKER, MQTT_PORT, 60)

# Setup serial
ser = serial.Serial(SERIAL_PORT, BAUD_RATE, timeout=1)
time.sleep(2)

print("📡 Monitoring INA sensors...")

while True:
    try:
        line = ser.readline().decode("utf-8", errors="ignore").strip()
        if not line:
            continue

        data = json.loads(line)

        if "ina1" in data:
            client.publish(TOPIC_INA1, json.dumps(data["ina1"]))
        if "ina2" in data:
            client.publish(TOPIC_INA2, json.dumps(data["ina2"]))

    except json.JSONDecodeError as e:
```

```
        print(f" JSON decode error: {e} | line: {line}")
    except Exception as e:
        print(f" General error: {e}")
        time.sleep(1)
```

Python Limit Switch Monitoring Script

This script reads the state of four limit switches from an Arduino (connected via /dev/arduino_limits) and publishes their status to MQTT for real-time system monitoring. The Arduino continuously outputs JSON-encoded boolean values for each switch (LS1 to LS4), which this script parses and republishes to the topic jetson/limits/ls. The script ensures that the latest switch states are made available to other system components, allowing immediate responses—such as halting or reversing motion when physical limits are reached. Robust error handling is implemented to tolerate malformed serial input and maintain continuous operation even under unstable conditions.

```
import serial
import json
import paho.mqtt.client as mqtt

# === Configuration ===
SERIAL_PORT = "/dev/arduino_limits"
BAUDRATE = 115200
MQTT_BROKER = "10.125.124.177"
MQTT_PORT = 1883
MQTT_TOPIC = "jetson/limits/ls"

# === MQTT Setup ===
client = mqtt.Client()
client.connect(MQTT_BROKER, MQTT_PORT, 60)
client.loop_start()

# === Serial Setup ===
ser = serial.Serial(SERIAL_PORT, BAUDRATE, timeout=1)

# === Track Previous State ===
last_state = {
    "LS1": None,
    "LS2": None,
    "LS3": None,
    "LS4": None
}

print("Listening on serial and publishing changes to MQTT...")

while True:
    try:
        line = ser.readline().decode("utf-8").strip()
```

```
        if not line:
            continue

        data = json.loads(line)
        changed = {}

        for key in ["LS1", "LS2", "LS3", "LS4"]:
            if key in data:
                if data[key] != last_state[key]:
                    changed[key] = data[key]
                    last_state[key] = data[key]

        if changed:
            client.publish(MQTT_TOPIC, json.dumps(last_state))
            print(f"Published to {MQTT_TOPIC}: {last_state}")

    except json.JSONDecodeError as e:
        print(f"JSON decode error: {e}")
    except Exception as e:
        print(f"Error: {e}")
```

Python EEG Command Handler Script

This script acts as the core logic layer for interpreting EEG-based control signals and enforcing physical safety constraints. It listens to two MQTT topics: jetson/eeg/command/servo for high-level EEG actions (e.g., "lift", "push", "left", "right") and jetson/limits/ls for limit switch states (LS1–LS4). When an EEG command is received, the script calculates the desired change in servo angle—while actively checking if movement is blocked due to a triggered limit switch. If a limit is hit, the servo is momentarily reversed to relieve pressure and movement is blocked until the switch is cleared. Servo positions are clamped within predefined ranges (servo1: 0–60°, servo2: 80–180°), and updated angles are published to jetson/command/servo for real-time actuation. The use of threading ensures that each MQTT message is handled promptly without blocking the main loop, and debug logs help verify timing and decision logic during testing.

```
import json
import time
import threading
import paho.mqtt.client as mqtt

# === MQTT CONFIG ===
BROKER = "10.125.124.177"
PORT = 1883
COMMAND_TOPIC = "jetson/eeg/command/servo"
LIMIT_TOPIC = "jetson/limits/ls"
OUTPUT_TOPIC = "jetson/command/servo"

# === Debug Mode ===
debug = True
```

```
# === Servo Limits ===
servo_limits = {
    "servo1": {"min": 0, "max": 60},     # Elbow
    "servo2": {"min": 80, "max": 180}    # Wrist
}

# === Initial Servo States ===
servo_angles = {
    "servo1": 30,   # Midpoint of 0–60
    "servo2": 145   # Midpoint of 180–80
}

movement_blocked = {
    "servo1": False,
    "servo2": False
}

# === Constants ===
STEP = 2.5
REVERSE_STEP = 2.5

client = mqtt.Client()

# === Limit Switch Handling ===
def on_limit_message(client, userdata, msg):
    start = time.perf_counter()
    try:
        payload = json.loads(msg.payload.decode("utf-8"))

        # --- Servo1 (Elbow): LS1 = max limit, LS2 = min limit ---
        ls1_triggered = payload.get("LS1", False)
        ls2_triggered = payload.get("LS2", False)
        if ls1_triggered:
            if not movement_blocked["servo1"]:
                movement_blocked["servo1"] = True
                # LS1 = max → reverse down
                servo_angles["servo1"] -= REVERSE_STEP
                servo_angles["servo1"] = max(servo_limits["servo1"]["min"],
servo_angles["servo1"])
                client.publish(OUTPUT_TOPIC, json.dumps({"servo1":
servo_angles["servo1"]}))
                if debug:
                    print(f"LS1 hit! Reversing servo1 down to
{servo_angles['servo1']}")
        elif ls2_triggered:
            if not movement_blocked["servo1"]:
                movement_blocked["servo1"] = True
```

```
                # LS2 = min → reverse up
                servo_angles["servo1"] += 5
                servo_angles["servo1"] = min(servo_limits["servo1"]["max"],
servo_angles["servo1"])
                client.publish(OUTPUT_TOPIC, json.dumps({"servo1":
servo_angles["servo1"]}))
                if debug:
                    print(f"LS2 hit! Reversing servo1 up to
{servo_angles['servo1']}")
        else:
            movement_blocked["servo1"] = False

        # --- Servo2 (Wrist): LS3 = max limit, LS4 = min limit ---
        ls3_triggered = payload.get("LS3", False)
        ls4_triggered = payload.get("LS4", False)
        if ls3_triggered:
            if not movement_blocked["servo2"]:
                movement_blocked["servo2"] = True
                # LS3 = max → reverse down
                servo_angles["servo2"] -= REVERSE_STEP
                servo_angles["servo2"] = max(servo_limits["servo2"]["min"],
servo_angles["servo2"])
                client.publish(OUTPUT_TOPIC, json.dumps({"servo2":
servo_angles["servo2"]}))
                if debug:
                    print(f"LS3 hit! Reversing servo2 down to
{servo_angles['servo2']}")
        elif ls4_triggered:
            if not movement_blocked["servo2"]:
                movement_blocked["servo2"] = True
                # LS4 = min → reverse up
                servo_angles["servo2"] += 5
                servo_angles["servo2"] = min(servo_limits["servo2"]["max"],
servo_angles["servo2"])
                client.publish(OUTPUT_TOPIC, json.dumps({"servo2":
servo_angles["servo2"]}))
                if debug:
                    print(f"LS4 hit! Reversing servo2 up to
{servo_angles['servo2']}")
        else:
            movement_blocked["servo2"] = False

    except Exception as e:
        print("Error handling limit switch message:", e)
    end = time.perf_counter()
    if debug:
        print(f"Limit handler time: {end - start:.4f}s")
```

```
# === Command Handling ===
def on_command_message(client, userdata, msg):
    start = time.perf_counter()
    try:
        payload = json.loads(msg.payload.decode("utf-8"))
        action = payload.get("action")

        if action == "push":
            servo = "servo1"
            delta = STEP
        elif action == "lift":
            servo = "servo1"
            delta = -STEP
        elif action == "right":
            servo = "servo2"
            delta = STEP
        elif action == "left":
            servo = "servo2"
            delta = -STEP
        else:
            return

        if movement_blocked[servo]:
            if debug:
                print(f"Movement blocked for {servo}, ignoring {action}")
            return

        # Update and clamp angle based on servo limits
        servo_angles[servo] += delta
        min_angle = servo_limits[servo]["min"]
        max_angle = servo_limits[servo]["max"]

        servo_angles[servo] = max(min_angle, min(max_angle,
servo_angles[servo]))

        client.publish(OUTPUT_TOPIC, json.dumps({servo: servo_angles[servo]}))
        if debug:
            print(f"Action '{action}' → {servo} = {servo_angles[servo]}")

    except Exception as e:
        print("Error handling servo command:", e)
    end = time.perf_counter()
    if debug:
        print(f"Command handler time: {end - start:.4f}s")

# === Threaded Wrapper ===
def threaded_callback(callback):
    def wrapper(client, userdata, msg):
```

```
        threading.Thread(target=callback, args=(client, userdata,
msg)).start()
    return wrapper

# === Setup MQTT ===
client.on_connect = lambda c, u, f, rc: (
    print("Connected with result code", rc),
    c.subscribe(COMMAND_TOPIC, qos=0),
    c.subscribe(LIMIT_TOPIC, qos=0)
)

client.message_callback_add(COMMAND_TOPIC,
threaded_callback(on_command_message))
client.message_callback_add(LIMIT_TOPIC, threaded_callback(on_limit_message))

client.connect(BROKER, PORT, 60)
client.loop_start()

# === Keep Main Thread Alive ===
try:
    while True:
        time.sleep(0.1)
except KeyboardInterrupt:
    print("Exiting...")
    client.loop_stop()
    client.disconnect()
```

Safe Operation

Assumptions for Safe Operation

The system can be safely operated under the following conditions:

1. **User Setup:**
    a. The user should be seated comfortably in a chair to ensure proper posture during operation.
    b. The user must wear the **Emotiv EpochX 14-channel headset** correctly, ensuring proper electrode contact for accurate data reading. *Test E-T08.*
    c. A nurse or trained operator should be present during the training sessions to assist the user and ensure safety.
2. **Training Sessions:**
    a. The system is designed for **15-minute training sessions** followed by **30-minute breaks**. Over time, training durations will adjust based on the user's fatigue level.
    b. Users must refrain from eating, drinking, or smoking during training. If the user feels fatigued, they should take a break and, if possible, resume the training once they feel comfortable. *Test E-T09.*
    c. The system will automatically stop if the user exceeds the safe operational range for the servos or other components, and a nurse or trained operator will have the physical **Emergency Stop button** if the user is unable to hold.
3. **Environmental Factors:**
    a. The system should be operated in a **well-ventilated environment** to prevent overheating of components, especially the servos and electronics.
    b. It must be ensured that the **emergency stop** (both software and physical) is clearly accessible and functional in the event of any failure or malfunction.
    c. Remove as many electronics from the vicinity to reduce noise and interference with the EEG headset.
4. **Health and Safety Monitoring:**
    a. The user should not operate the system if feeling unwell or fatigued.
    b. The user should not operate the system if they are alone and/or without a trained professional.
    c. There should be clear instructions and training on **how to operate the system safely**, particularly for both the user and the nurse, which are included above and will be present in a **user manual**.

Safety Features

The system is equipped with several built-in safety features to ensure the safe operation of both the user and the hardware:

1. **Range Limitation and Limit Switches for Servos:**
    a. **Servos** used in the system are range-limited to prevent excessive motion that could lead to injury or damage. If a servo attempts to exceed its range, **limit switches** will activate, signalling the system to stop the movement and reverse by 2.5 degrees. This ensures the system doesn't push beyond safe mechanical boundaries.
2. **Over-Current Protection:**
    a. **Over-current protection** is implemented on both the servos and the entire system. The servos themselves are equipped with their own **over-current protection circuits** to prevent overheating or damage from excessive current draw. Additionally, **current sensors** are employed to measure and monitor the overall system's current usage, further preventing any possible electrical failure or fire hazard due to an overload.
3. **Emergency Stop (E-Stop):**
    a. The system includes both a **software-based emergency stop** accessible via the Node-RED dashboard and a **physical emergency stop button**. The software-based emergency stop will immediately halt all servo operations if triggered, while the physical button provides an additional layer of safety in case of an emergency or malfunction and has the same function as the software e-stop.
4. **Battery Safety:**
    a. The system runs on **two batteries in parallel** to ensure reliable power supply. This configuration helps prevent sudden power loss and ensures that the system will continue to operate safely if one battery is depleted or fails. The batteries last for 2 hours of constant use, and the headset lasts an hour when fully charged. The batteries are also designed with built-in safety mechanisms to prevent overheating and overcharging.
5. **Human-Machine Interaction (HMI) Safety:**
    a. The user will interact with the system via a **laptop running training software**, with real-time data streaming from the **Emotiv EpochX headset**. The nurse will be responsible for ensuring the user's well-being and intervening if any issues arise during the training session and mounting the arm to the user.
    b. The nurse will be trained to recognize signs of user discomfort or fatigue and will be responsible for ensuring that the user follows the recommended training schedules, including taking appropriate breaks.

6. **Software Safety Monitoring:**
    a. The system software provides **real-time feedback** on both the hardware (servos, battery status) and user data (via the EEG headset). If any safety limits are exceeded (e.g., servo range, over-current, or system failure), the software will trigger appropriate actions, including stopping movement.

Justification of Safety Features

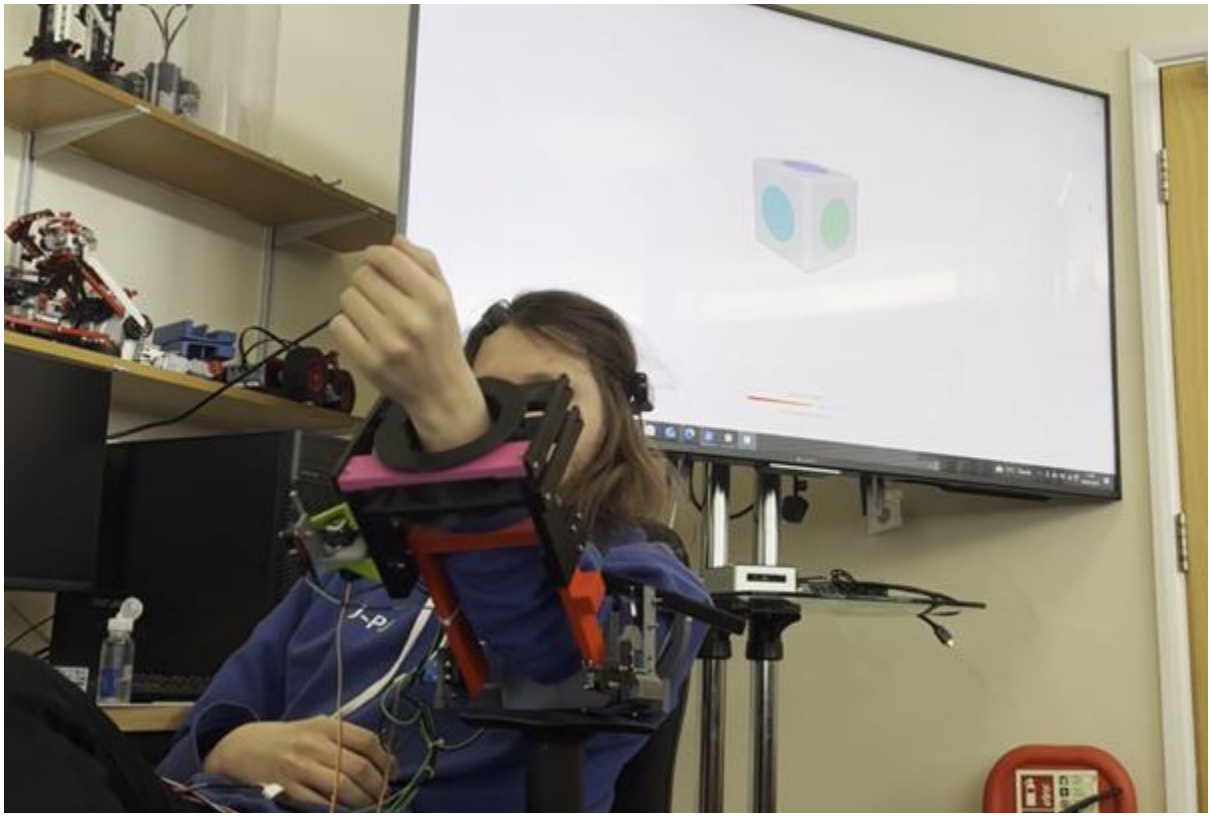
The safety features integrated into the system are justified by the following reasons:

1. **Servo Range Limitation and Over-Current Protection:**
    a. Ensuring the servos remain within a predefined range prevents mechanical damage and minimizes the risk of injury to the user. The over-current protection ensures that the system operates within safe electrical limits, preventing damage from excess current and reducing the risk of electrical fires.
2. **Emergency Stop Mechanisms:**
    a. Both the physical and software emergency stops are vital to prevent further system operation in case of a fault, such as servo failure, overheating, or an unexpected hardware malfunction. The availability of two stop mechanisms (software and physical) adds redundancy to the safety system, ensuring the operator or user can stop the system quickly in case of any hazard.
3. **Battery Safety:**
    a. Using two parallel batteries ensures that the system remains operational even in the event of battery failure. The additional battery safety mechanisms protect the system from electrical hazards and ensure that it can be safely recharged without risk of overcharging or overheating.
4. **Human-Machine Interface (HMI) Safety:**
    a. Having a nurse trained in the safe operation of the system ensures that there is someone who can monitor the user and intervene if necessary. The nurse's role in guiding the user through training sessions ensures that the user remains within the system's safe operating parameters.
5. **System Monitoring and Feedback:**
    a. Real-time monitoring of the system's performance, user data, and hardware status helps ensure that any issues are immediately detected and addressed. By providing alerts and feedback to the nurse, the system helps maintain a safe and controlled training environment.

Overall, these safety features are designed to ensure the safety of both the user and the system, promoting a controlled and safe training experience while minimizing the risk of injury, equipment failure, or system malfunction. This structure addresses the necessary assumptions and safety features, providing a clear and justified explanation for the measures taken to ensure safe operation.

User Manual

This manual provides step-by-step instructions for safely operating the robotic arm system with the Emotiv EEG headset. It includes setup, operation, and troubleshooting guidance. A nurse, carer or family member with the appropriate training to use the device and software.



1. Safe Operation

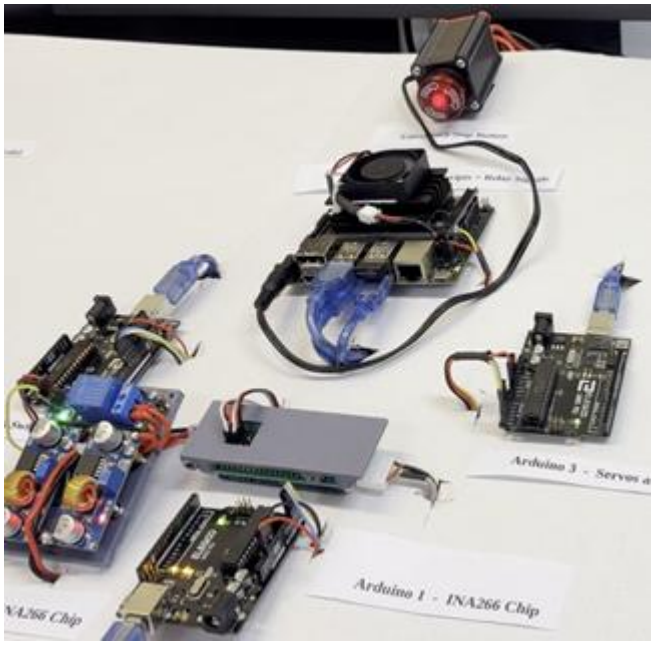
Assumptions for Safe Operation

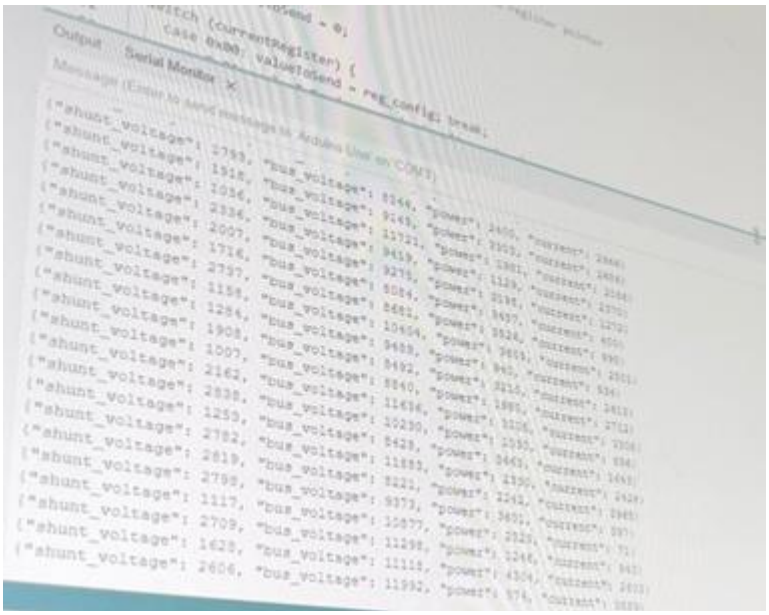
- The system is designed to be operated only when the user is seated comfortably in a chair, with a properly fitted **Emotiv EpochX 14-channel headset**.
- A trained nurse will be present during training to ensure the user is positioned correctly and safe throughout the session.
- **Training Sessions**: The system operates with 15-minute training intervals followed by 30-minute breaks. The user should avoid eating, drinking, or smoking during the training. If the user feels fatigued, the nurse should guide them to take a break.

Safety Features

- **Emergency Stop (E-Stop)**: Both software-based and physical emergency stop buttons are available.
- **Servo Range Limitation and Over-Current Protection**: Prevents damage or injury by limiting servo movement and monitoring electrical usage.
- **Battery Safety**: Dual parallel batteries with built-in safety mechanisms for preventing overheating and overcharging.

- **Human-Machine Interface (HMI) Safety**: Nurse supervision ensures safe operation and intervention during the training.
- **System Monitoring and Feedback**: Real-time system monitoring with alerts for abnormal conditions.





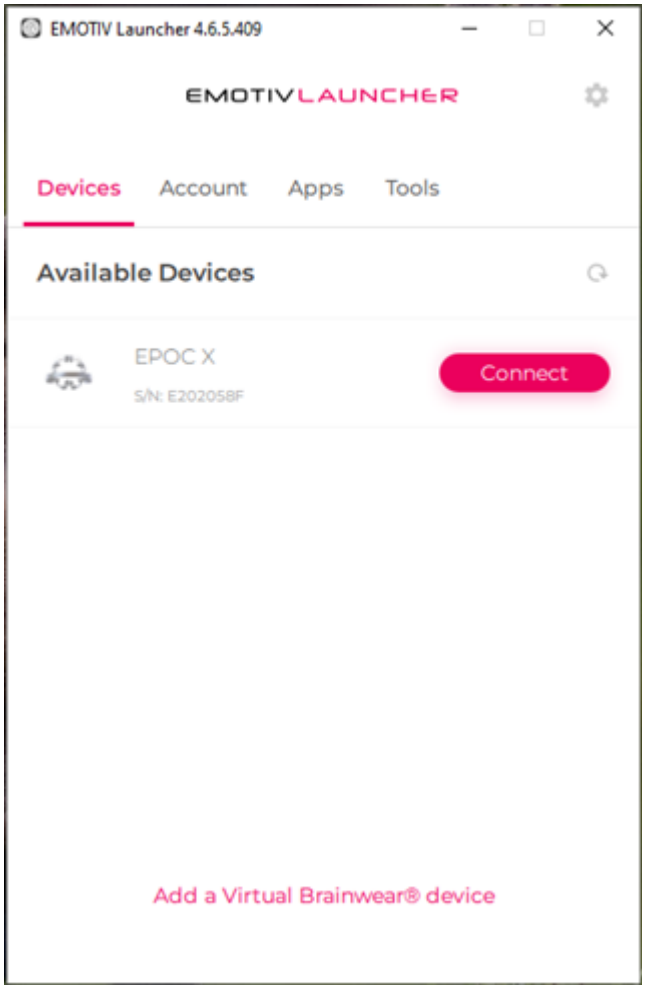
2. Setting Up the System

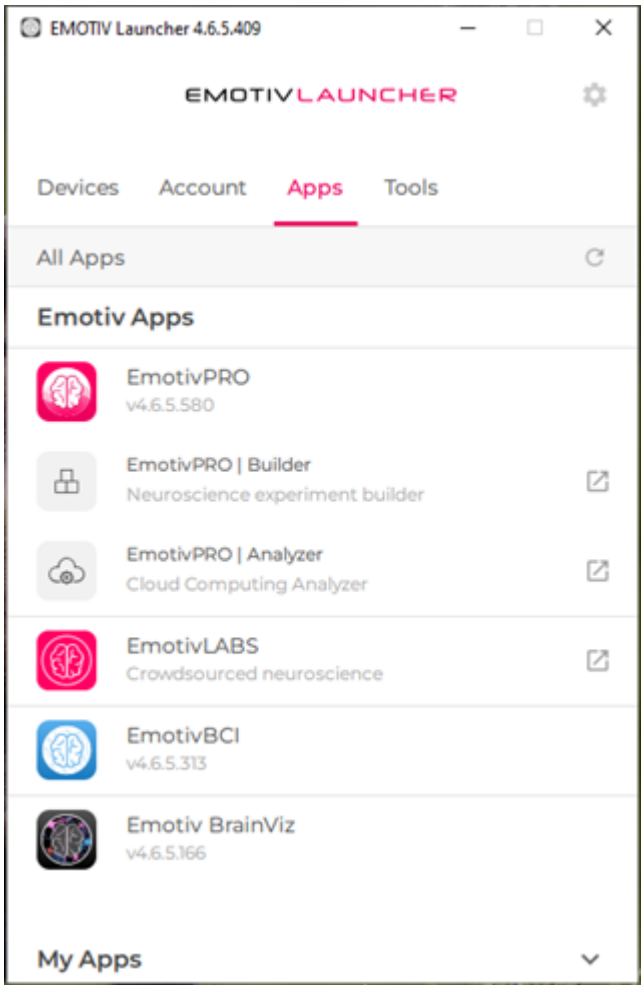
Step 1: Mounting the Robotic Arm

1. **Adjusting Velcro Straps**:
    a. The nurse will help the user strap their arm into the robotic arm. The adjustable Velcro straps are designed to rest on foam sections, ensuring the user's arm is comfortably supported.
    b. The **bicep section** should be placed **above the elbow**, and the **hand should be placed through the arm's circular section**. This setup ensures the arm is positioned correctly for optimal movement and safety during training.

Step 2: Preparing the Emotiv EpochX Headset

1. **Positioning the Headset**: Ensure that the headset is properly worn by the user. The electrodes should have **good contact with the skin** to ensure accurate readings.
2. **Start the Emotiv Software**:
   a. Launch the **Emotiv software** on the laptop.
   b. Ensure the software indicates that the connections are **"on"** and that the headset is ready.





Step 3:

Setting Up Node-RED

1. **Start Node-RED**:
   a. On the laptop, **open PowerShell** and type node-red and press **Enter**.
   b. This will start the Node-RED server, allowing communication between the software and the robot.

2. **Launch Docker and Start MQTT Container**:
    a. After Node-RED is running, launch **Docker** on the system.
    b. Find the **MQTT container** and click the **"Run"** button to initialize the MQTT server.
3. **Connect the Jetson Nano**:
    a. Plug in the **Jetson Nano** to the laptop. The **auto-script** will run automatically, booting up necessary scripts (e.g., main, driver, MQTT).
    b. Once the setup is complete, Node-RED will receive a **"ready" response** confirming that the system is operational.

3. Starting the Training

1. **Verification**:
    a. Confirm that the **Emotiv software** is displaying correct EEG data and that the system is communicating properly with Node-RED.
2. **Begin Training**:
    a. Once all systems are confirmed as ready, the **nurse** will start the training session. The user will begin the **15-minute training session**, during which the robotic arm will assist in training based on EEG data.
    b. The system will monitor for any potential safety risks (e.g., servo over-range or excessive current draw) and stop operations if necessary.

4. Troubleshooting & Debugging

If any issues arise during setup or operation, follow these steps to troubleshoot:

1. **Emotiv Software Issues**:
    a. **No Connection**: Ensure the **headset electrodes** have proper contact with the user's skin. Check the USB connection between the headset and the laptop.
    b. **Test Connection**: Open the Emotiv software and check if the **connections display as "on"**. If not, reconnect the headset and verify that the software detects the device.
2. **Node-RED Issues**:
    a. **Node-RED Not Starting**: If Node-RED does not start, open PowerShell and type node-red again. If there's an error, check the script for issues or restart the laptop.
    b. **Communication Issues**: If Node-RED doesn't receive a "ready" response, ensure the **Jetson Nano** is properly connected to the laptop and powered.
3. **Servo Issues**:
    a. If the servos are not responding, check for **over-current conditions** or any **range limits** being exceeded. The system will automatically stop the servo if these limits are triggered. If needed, press the **emergency stop** button.
4. **General Hardware Issues**:
    a. **Power Issues**: If the system is not receiving power, ensure both **batteries are charged** and connected properly. If one battery fails, the system should continue operating using the second battery.
    b. **Emergency Stop Triggered**: If the emergency stop is activated, the system will halt all servo operations. Ensure that the user is safely seated and that the system can be reset.

5. Safety Precautions

1. **Fatigue**:
    a. If the user feels fatigued, they should immediately stop the training and take a break. The system is designed to adjust training based on the user's level of fatigue.
2. **Emergency Stop**:
    a. If at any point the user or the nurse feels that the system is malfunctioning, the **emergency stop button** can be pressed to stop the system immediately.
3. **Environmental Considerations**:
    a. Ensure the system is operating in a **well-ventilated space** to prevent overheating.
    b. Keep electronics away from the EEG headset to avoid interference.

Further Work

Unfinished Features

INA219 Current Sensor Chip

Description of Sub-Component

Midway through the development of the arm, it was decided that there would have to be some sort of safety component to stop the servo motors from bending the arm too far. This could be detected through the current being supplied to the motor as it would be much higher if it is having to work against the strength of the arm.

This then prompted a search for devices and methods for detecting the current with an ability to simply integrate it into a safety mechanism. The best solution for this problem was the INA219 I2C chip that can read a range of different values including the current and feed them back to an I2C compatible device using serial. This is perfect when paired with an Arduino to make a closed circuit that would shut off a relay when the current is too high.

The relay is a 5V model with a coil and switch. The supply is then connected to a usually off or usually on side to decide how the relay should be operated. As this is a safety feature, the relay should default to off which means the circuit was designed in this way.

As well as this, 2 INA219 chips will be required for each of the servos so that each is being monitored individually. With each of these in place it would be impossible for a situation to arise where one servo is drawing too much current, but the relay isn't switched because a joined current monitor doesn't sense above the threshold.

Functional Requirements

The functional requirements that this sub-component of the system are as follows:

- The Component must be able to monitor the current of each servo live to a high degree of accuracy.
- It must be able to shut off all of the power to the servos is a certain threshold is reached on one or both of the servos.
- It must send the current as well as any other readings from the chip as a serial JSON output.
- It must all be self-contained and not dependent on any other aspect of the entire system.
- It must have a powered off fail state.
- Must be in a small container that doesn't encroach on the original design of the arm.
- It needs to be easily connectable in series as an "in-between" component of the circuit.

- Each component of the circuit should be individually testable and removable from the main dataflow.

Development of Component

Development started on this by using the INA219 chip and an Arduino to see if the I2C connection was operational. Once this was accomplished and it was giving an accurate reading, the if-statement was written into the code to check for an appropriate current to limit the circuit. This was tested at a lower threshold to show that the code worked as it was only turning the onboard LED to show whether the value was over it. This was set at 700 for the fan motor tests on the Arduino as the current regularly fluctuated above and below this value.

After this, a piece of code was written to test that the relay functioned as intended and won't turn anything on that is meant to be off. It was critical that this was tested as this was intended as a safety feature and wouldn't be validated as such with faulty components.

To add the second INA219 chip to the circuit, the address had to be connected by a solder bridge to differentiate the second chip from the existing one. The code was then altered to contain arrays, and the Json was also updated to reference the chips by their unique address (0x40 & 0x44).

During testing with two relays and two INA219 chips, the I2C connection stopped working after the relays were turned off. This could have been due to a surge in current as the relay has an electromagnetic field which could have caused this. This can be resolved by using diodes to prevent the current from surging where it shouldn't as well as testing the component using an externally powered system to put through the current sensor.

Programming Approach

Due to the nature of this component and how it integrates with the project, it was decided to write functions that achieved each task so that each one could be turned on or off for each test. This also means that if the team didn't want the relays to trigger for a particular test, it is possible to only log this to the console and move on from that particular function.

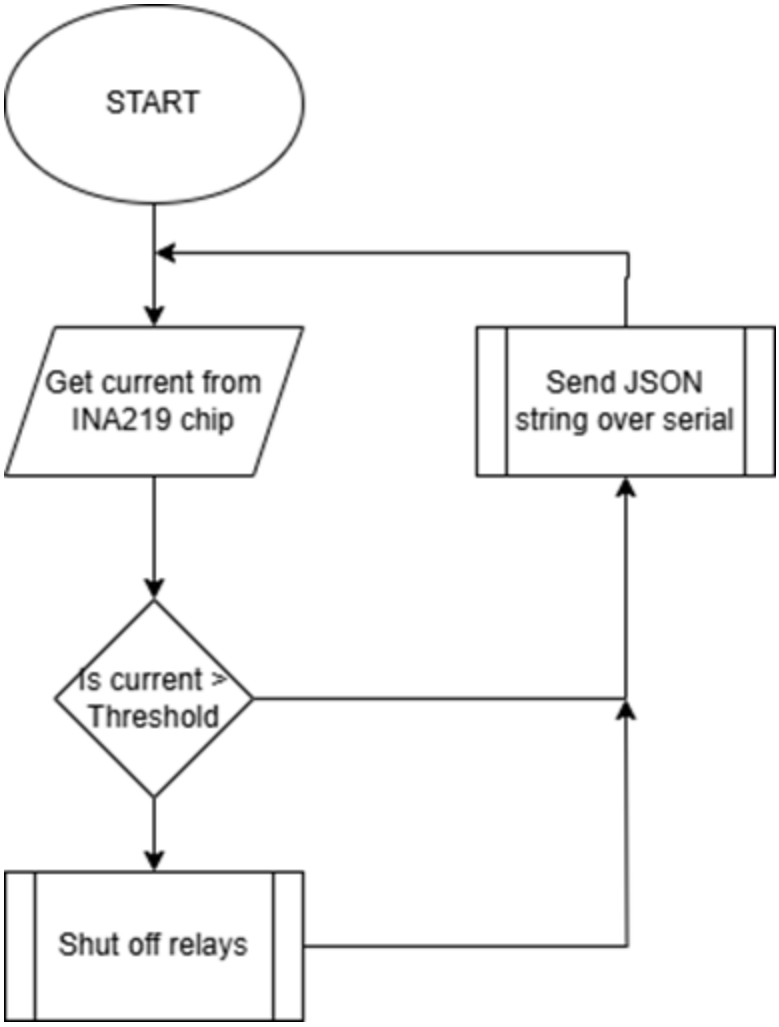


Figure 35 - INA Data Flow

The flowchart above shows a high-level overview of the Arduino code. It operates in a loop that is only terminated by resetting or unplugging the Arduino. This is as the program on this Arduino is treated as an essential safety feature and if there is power to the system, it should be operational. The arrow going down from the decision in the flowchart represents the True outcome which would result in the relays being turned off indefinitely until the Arduino is reset. The other outcome is the "False" outcome where the Json is sent over serial to the central controller. This happens during either outcome and is achieved by taking the inputs from the current sensor as variables and combining them all in a Json string. This is standardised with the rest of the Arduinos using serial communication to ensure that the jetson that is running the central control program doesn't have to make sense of multiple different Json formats.

Discontinuation of Sub-Component

While the component itself was built and tested, in the end this wasn't added to the final prototype due to adequate safety measures being in place for the testing of the system. However, if this was to be launched to the public, it would be a useful step towards not requiring supervision for the system to operate safely. There were INA266 chips integrated into the system due to their higher current capabilities and ability to detect voltage. These were instead used to log the status of various components throughout the system.

The component was tested separately using a fan motor being ran through an independent power source that passed through the relays connected to the Arduino. During this testing, the system was able to automatically detect spikes in the current due to resistance that was being exerted on the motor proving that the safety system could be added to the project as a whole at any point in time without interfering with the components that are already in place.

Simulation Using ROS2 & Gazebo

Initially, there was an intention to simulate the physical system using Gazebo to allow for training of the Emotiv commands. This would have consisted of python scripts which would have received commands from the EEG headset and used these to manipulate a simulated system.

The main components of the Gazebo Simulation would have been:

- Robot Description
  - Describes the Links, joints, sensors & physical properties.
  - URDF or SDF models would have been used to describe the physical collisions of the system as well as the visuals in the simulation.
- ROS2 Nodes
  - These have the ability to control the manipulation of robot arms that are similar to the system developed in this project.
  - Can publish or subscribe to sensor data
  - Are also able to provide high level logic such as planners or behaviour trees
- World Files
  - While these wouldn't have been useful to this project in particular. There is the ability to create a world file that a robot can either traverse or interact with using sensors and other methods of obstacle detection.
- ROS to Gazebo Communication Bridge
  - This ensures that Gazebo publishes topics that ROS2 nodes can subscribe to as well as ROS2 communicating back
- Launch files
  - These are written in python for ROS2.
  - Are often used to launch gazebo as well as any robot description files and controllers that are paired with it.
  - This is the part of the simulation that failed to work as the machines in the lab had recently been updated to ROS2 and the package which launched Gazebo wasn't installed correctly causing it to crash on start-up.

Before the development of this simulation was discontinued, the description of the robot was using primitive shapes and joints to show a rough outline of what the system would look like.

Before the controllers were written, it was essential that the launch files were able to launch gazebo with the robot present in the simulation. Once it was obvious that this wasn't going to work, this aspect of the project was discontinued.

In retrospect, this element of development could have benefited the team. It would have allowed the team working on the EEG headset to send their commands to a tangible system that reacted to what was being sent. It would have also helped to understand how the physical system should react to the commands that were being sent. If the reaction of the simulated system to a command is too subtle or too dramatic, this could have been tweaked without the risk of harm or damage to the equipment that comes with the physical testing of the system. It also could have shown potential issues that could become apparent once the system is built before it is close to completion allowing extra time to resolve these issues.

System Improvements

While the prototype successfully demonstrated EEG-controlled actuation for rehabilitation, several improvements could enhance reliability, usability, and scalability in future iterations. Transitioning from USB-connected Arduinos to a custom PCB or integrated microcontroller board would reduce wiring complexity and improve robustness. Incorporating onboard data logging and feedback (e.g. encoder feedback from servos) would allow finer motion control and better fault detection. Software-wise, migrating to a unified interface—either through a more robust ROS2 implementation or a custom web-based UI—could streamline monitoring and user interaction. Finally, upgrading the mechanical design to include stronger, lighter materials and more precise joints would further align the system with real-world rehabilitation needs, paving the way for clinical testing and refinement.

References

Emotiv. (n.d.) Pro license updates: New features and licensing options. Available at: https://www.emotiv.com/blogs/press/pro-license-updates-new-features-and-licensing-options (Accessed: 1/05/2025).

Nuwer, M. R., Comi, G., Emerson, R., Fuglsang-Frederiksen, A., Guérit, J., Hinrichs, H., Ikeda, A., Jose C. Luccas, F. & Rappelsburger, P. (1998) IFCN standards for digital recording of clinical EEG. *Electroencephalography and Clinical Neurophysiology,* 106 (3), 259–261.

Casson, A. J. (2019) Wearable EEG and beyond. *Biomedical Engineering Letters*, 9 (1), 53–71.

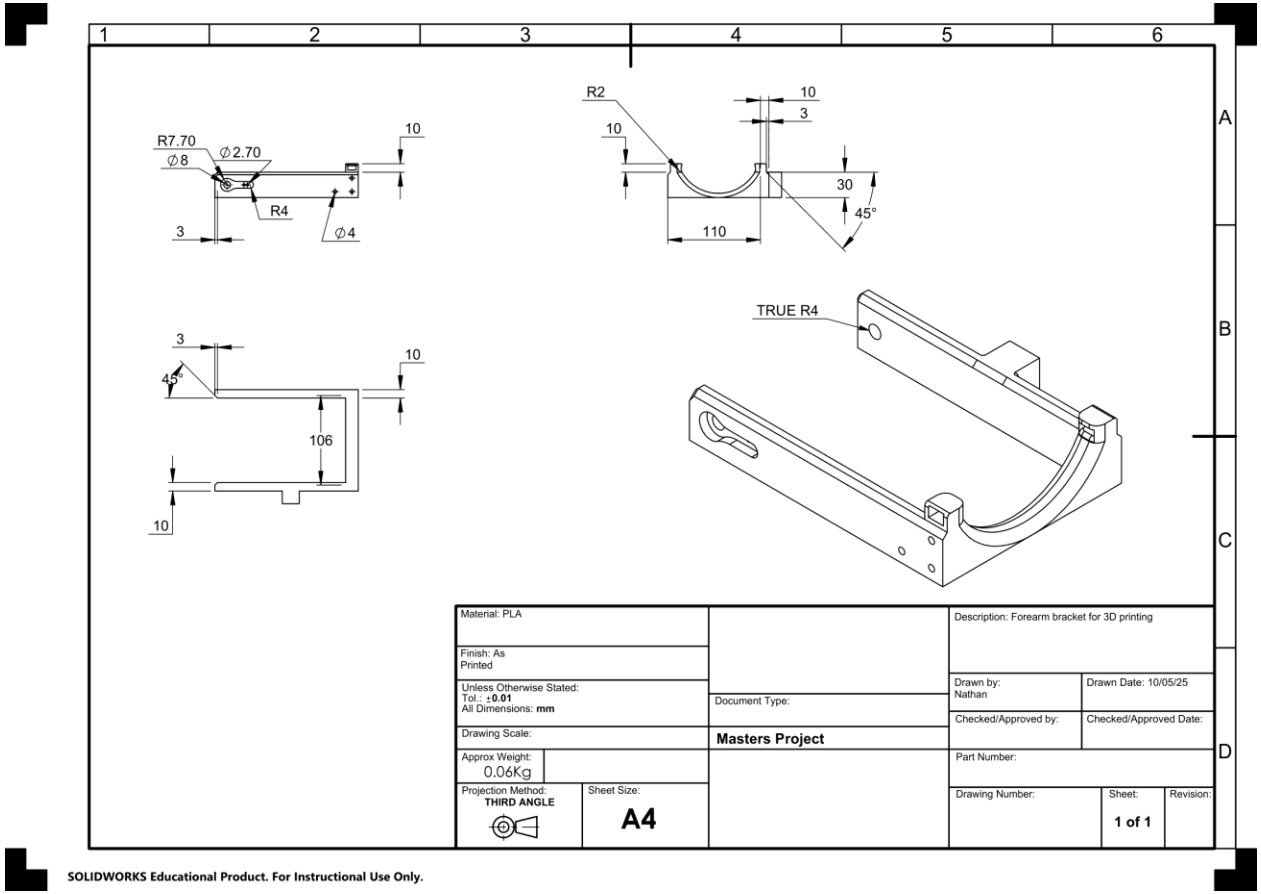
Bibliography

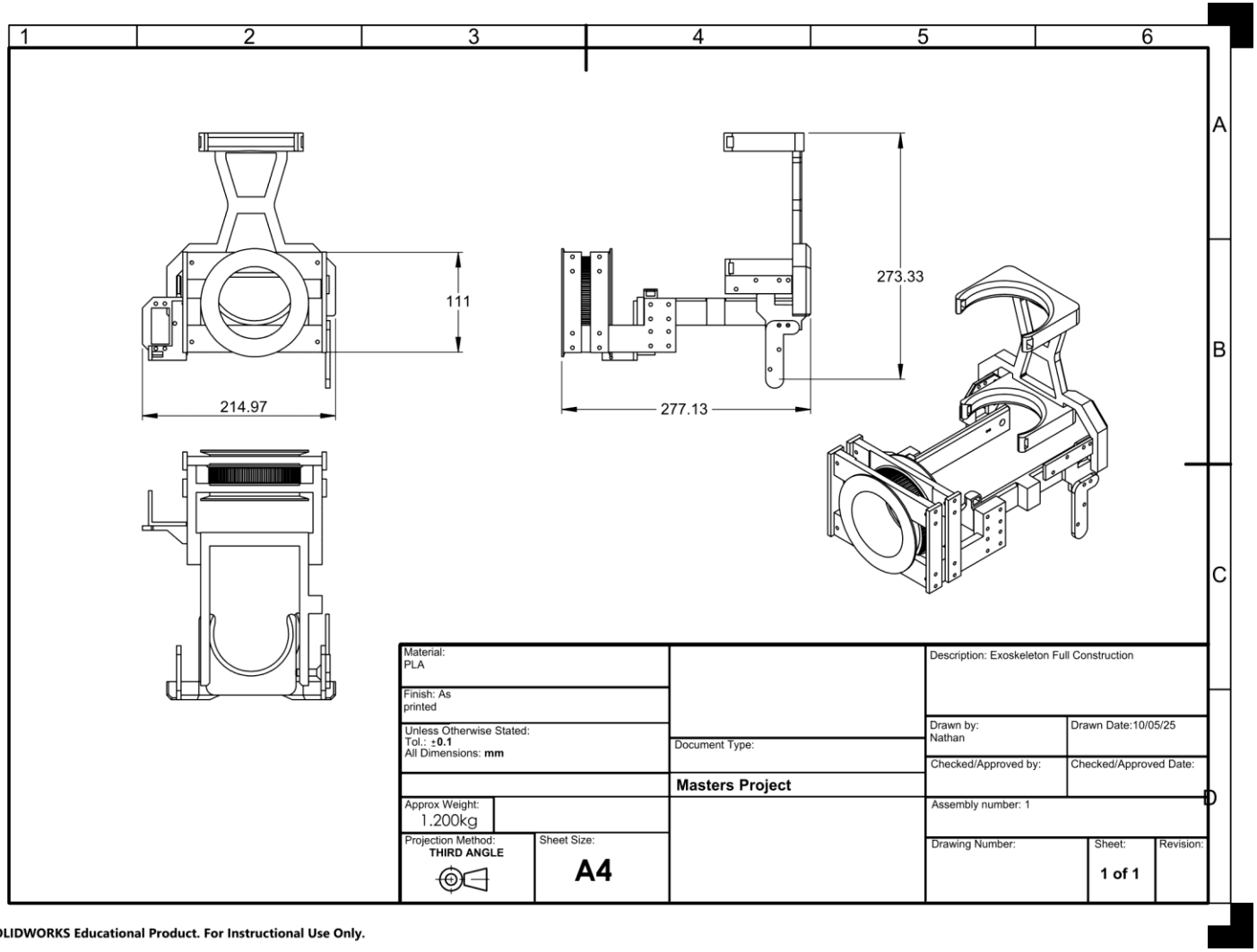
Node-RED. (n.d.) *Documentation*. Available at: https://nodered.org/docs/ (Accessed: 10/10/2024).
Eclipse Mosquitto. (n.d.) *Mosquitto – An Open Source MQTT Broker*. Available at: https://mosquitto.org/ (Accessed: 12/10/2024).
Docker. (n.d.) *Docker Documentation*. Available at: https://docs.docker.com/ (Accessed: 14/10/2024).
Visual Studio Code. (n.d.) *Remote - SSH Extension*. Available at: https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.remote-ssh (Accessed: 16/10/2024).
Adafruit. (n.d.) *16-Channel 12-bit PWM/Servo Driver - I2C interface - PCA9685*. Available at: https://learn.adafruit.com/16-channel-pwm-servo-driver (Accessed: 18/10/2024).
NVIDIA. (n.d.) *Jetson Nano Developer Kit User Guide*. Available at: https://developer.nvidia.com/embedded/jetson-nano-developer-kit (Accessed: 22/10/2024).
Arduino. (n.d.) *Arduino IDE Documentation*. Available at: https://docs.arduino.cc/software/ide-v2 (Accessed: 24/10/2024).
Python Software Foundation. (n.d.) *Python Documentation*. Available at: https://docs.python.org/3/ (Accessed: 26/10/2024).
Python Package Index. (n.d.) *PyPI – the Python Package Index*. Available at: https://pypi.org/ (Accessed: 28/10/2024).

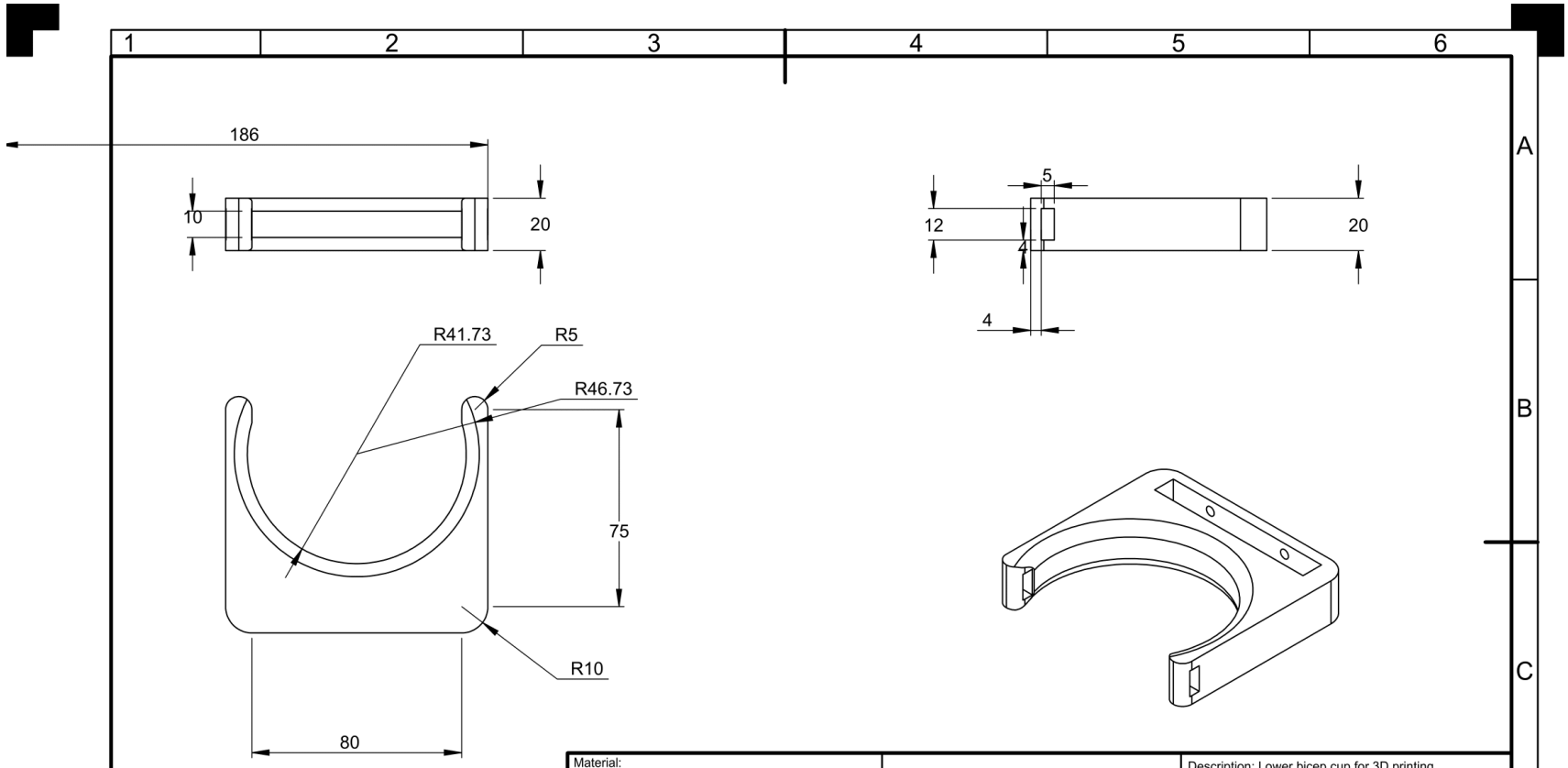
Rob Tillaart. (n.d.) *INA226 Library Documentation*. Available at: https://github.com/RobTillaart/INA226 (Accessed: 20/2/2024).

Appendix

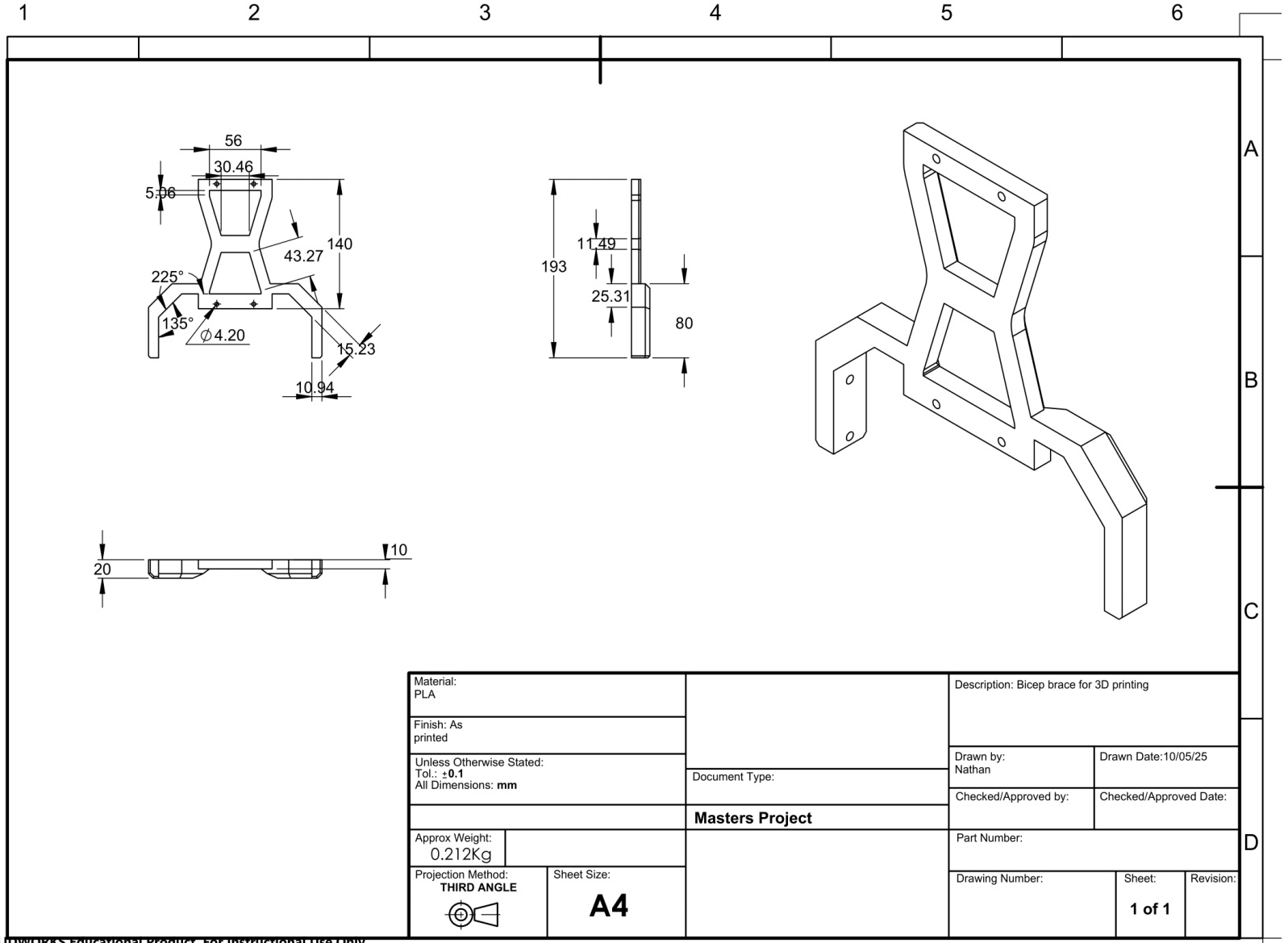
Appendix A - Engineering Drawings

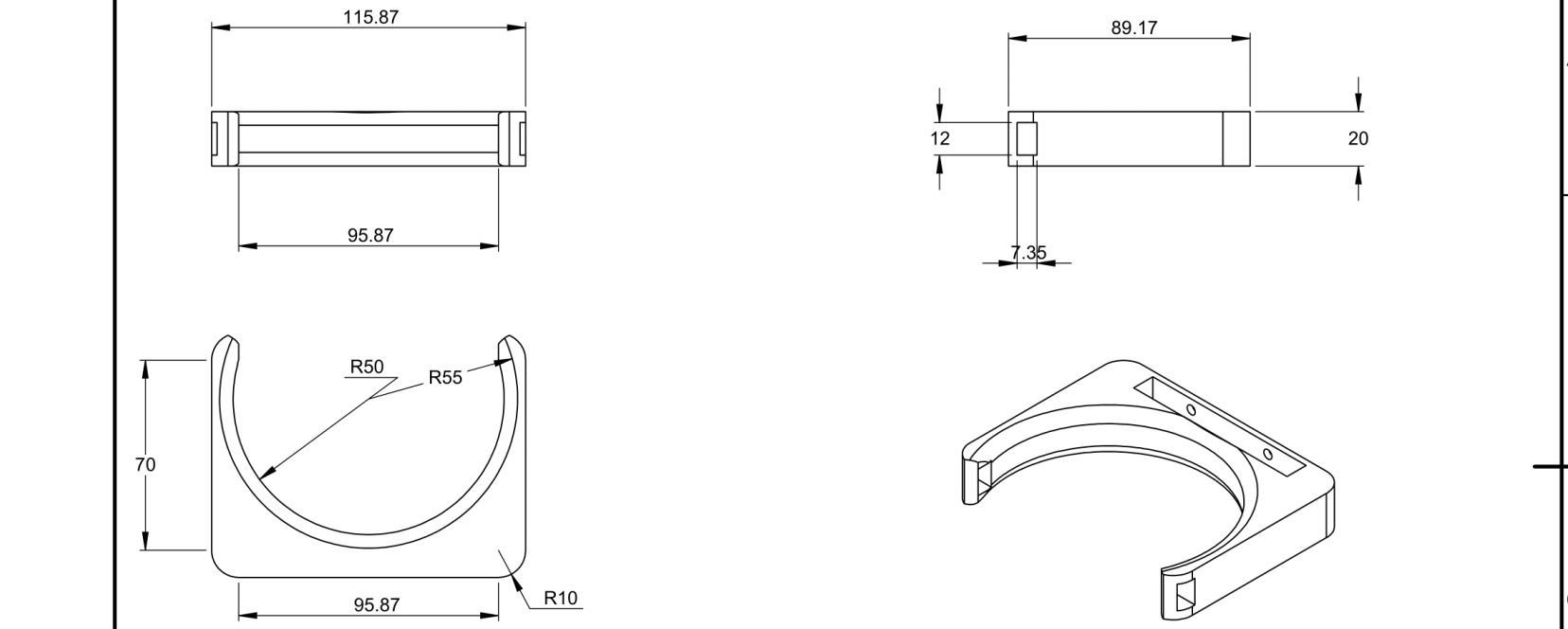



1
2
3
4
5
6
A
B
C
D
111
214.97
273.33
277.13
Material:
PLA
Finish: As
printed
Unless Otherwise Stated:
Tol.: ±0.1
All Dimensions: mm
Approx Weight:
1.200kg
Projection Method:
THIRD ANGLE
Sheet Size:
A4
Document Type:
Masters Project
Description: Exoskeleton Full Construction
Drawn by:
Nathan
Drawn Date:10/05/25
Checked/Approved by:
Checked/Approved Date:
Assembly number: 1
Drawing Number:
Sheet:
1 of 1
Revision:
SOLIDWORKS Educational Product. For Instructional Use Only.

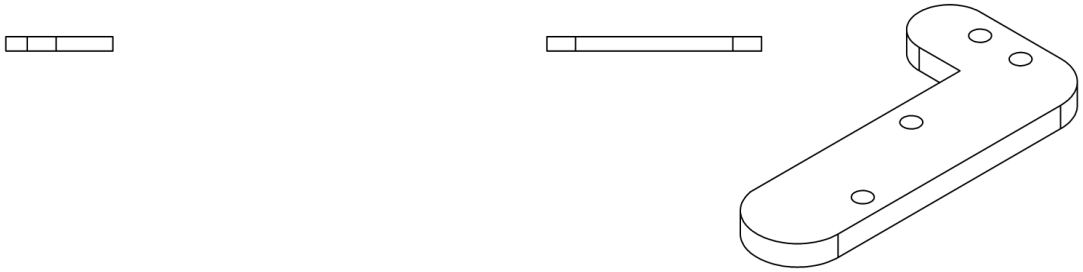


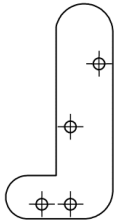
Material:		Description: Lower bicep cup for 3D printing
Finish:		
Unless Otherwise Stated: Tol.: ±0.1 All Dimensions: mm	Document Type:	Drawn by: Nathan / Drawn Date: 10/05/25
	Masters Project	Checked/Approved by: / Checked/Approved Date:
Approx Weight: 0.039Kg		Part Number:
Projection Method: THIRD ANGLE	Sheet Size: A4	Drawing Number: / Sheet: 1 of 1 / Revision:



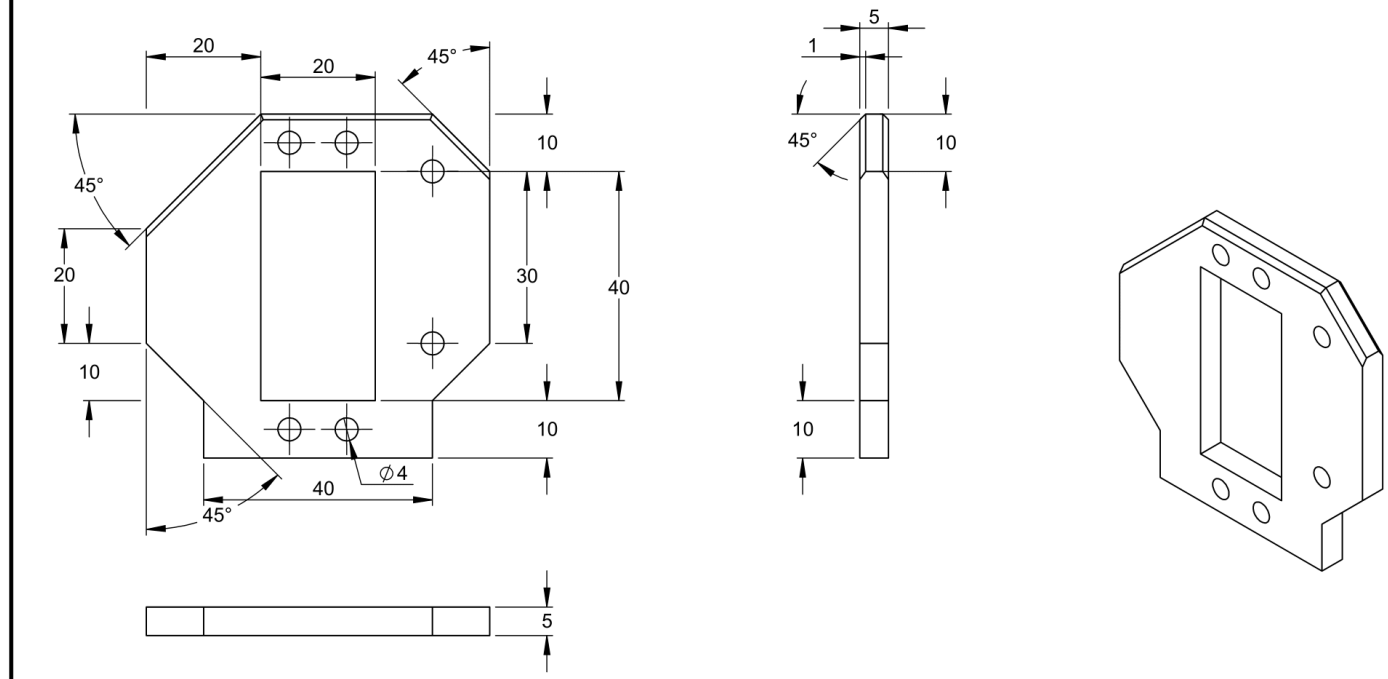


Material: PLA		Description: Lower bicep cup for 3D printing	
Finish: As printed			
Unless Otherwise Stated: Tol.: ±0.1 All Dimensions: mm	Document Type:	Drawn by: Nathan	Drawn Date: 10/05/25
		Checked/Approved by:	Checked/Approved Date:
	Masters Project		
Approx Weight: 0.040Kg		Part Number:	
Projection Method: **THIRD ANGLE**	Sheet Size: **A4**	Drawing Number:	Sheet: **1 of 1** / Revision:

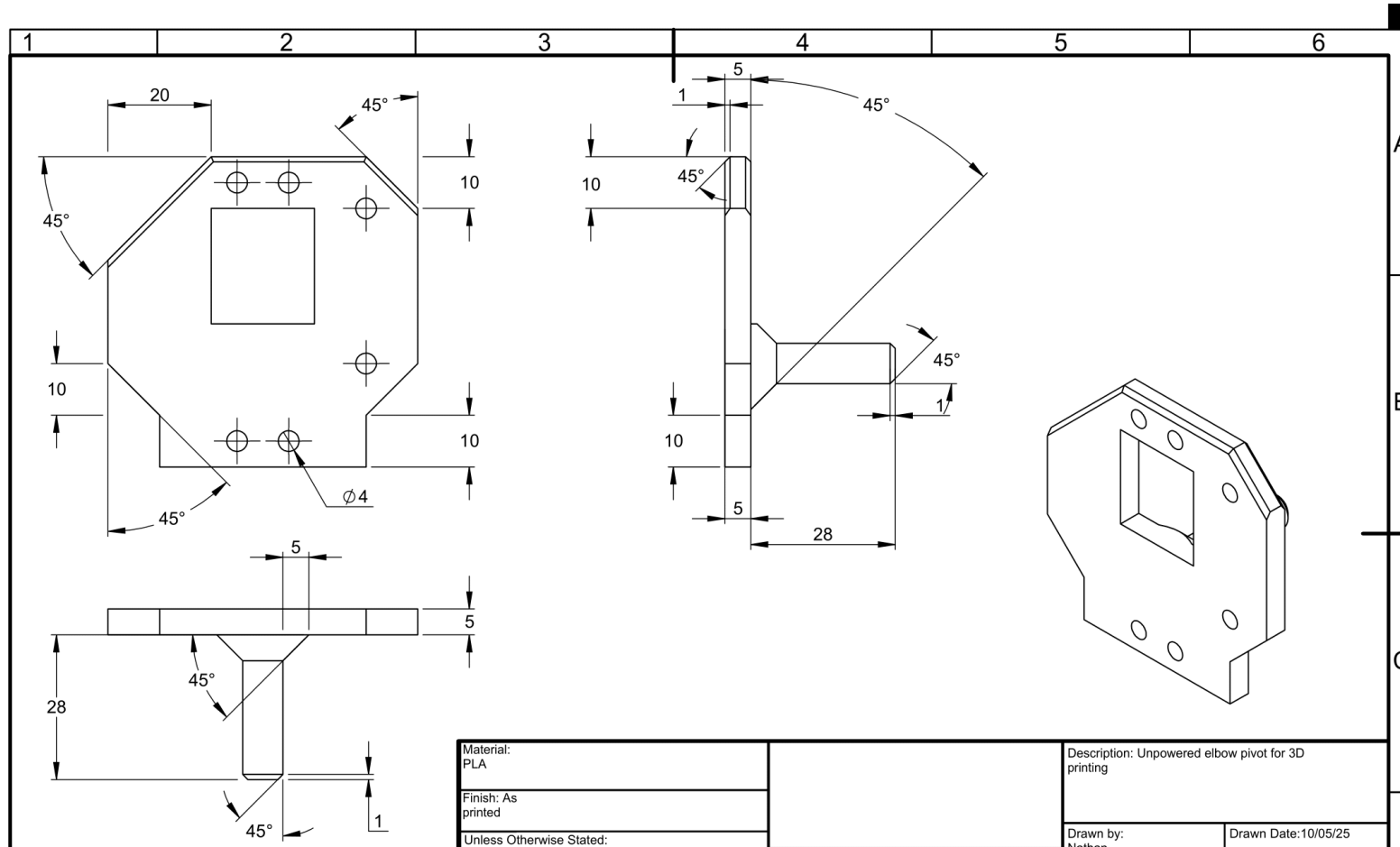


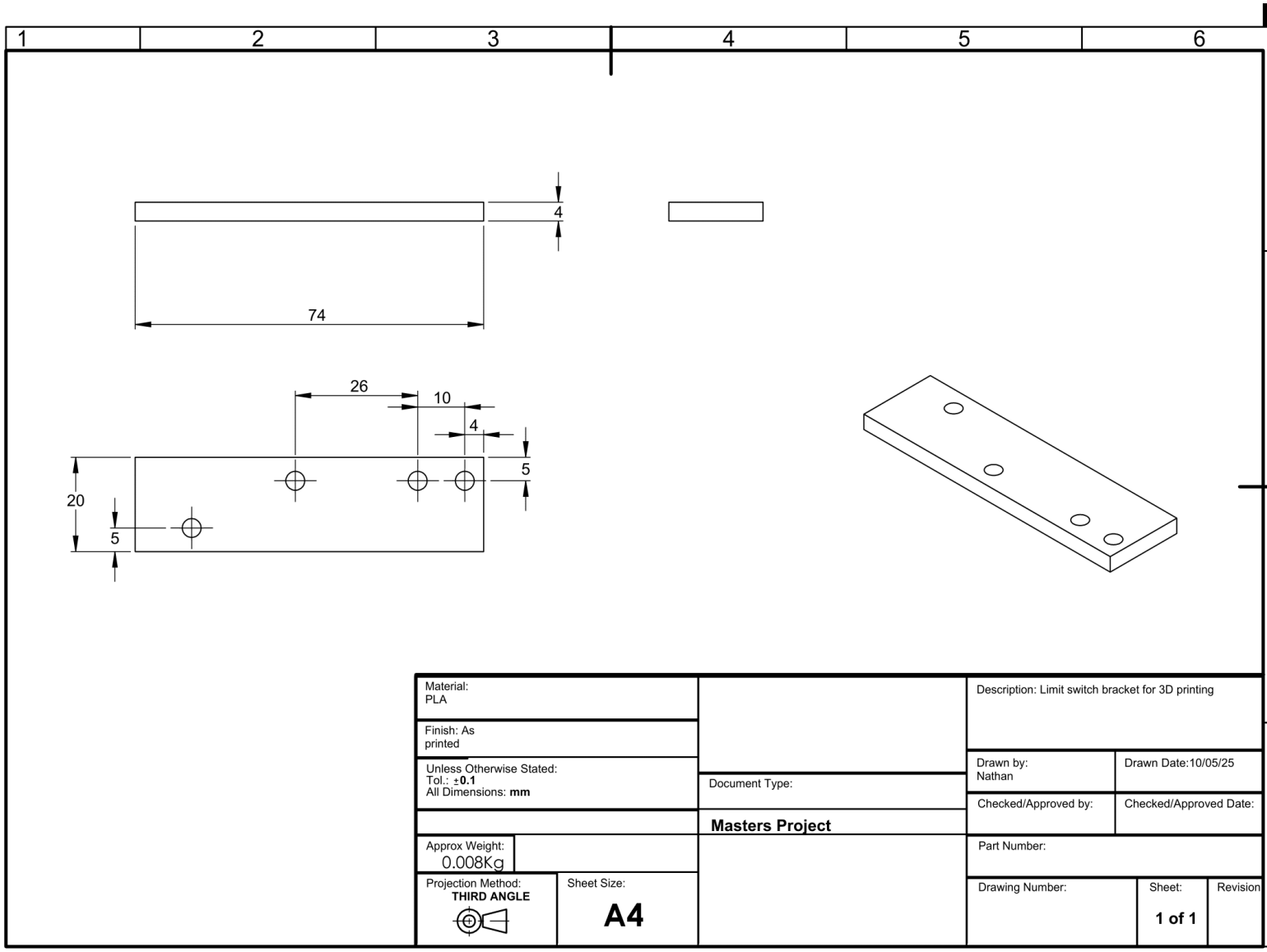


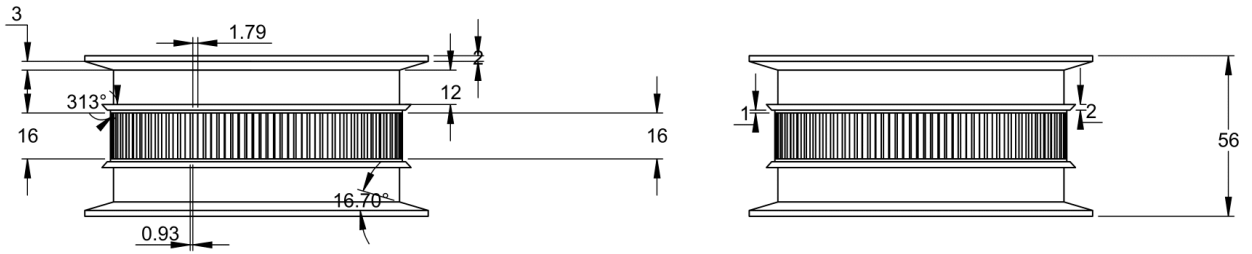
Material:		This drawing and any information or descriptive material set out on it are the confidential and copyright property of ***Company Name***. © and MUST NOT BE DISCLOSED, COPIED, LOANED in whole or part or used for any purpose without the written permission of ***Company Name***.	Description:	
Finish:			Drawn by:	Drawn Date:
Unless Otherwise Stated: Linear Tol.: ±**0.2**, Angular Tol.: **0°15'** Surface Finish: **0.8µm** All Dimensions: **mm**		Document Type:	Checked/Approved by:	Checked/Approved Date:
Drawing Scale: **1:2**		**Company Confidential**		
Approx Weight: Kg	Drawing Produced In Accordance With: **BS8888**	Legal Owner: **Company Logo** Company Name, Address and Tel. No e-mail address	Part Number:	
Projection Method: **THIRD ANGLE**	Sheet Size: **A4**		Drawing Number:	Sheet: **1 of 1** / Revision:

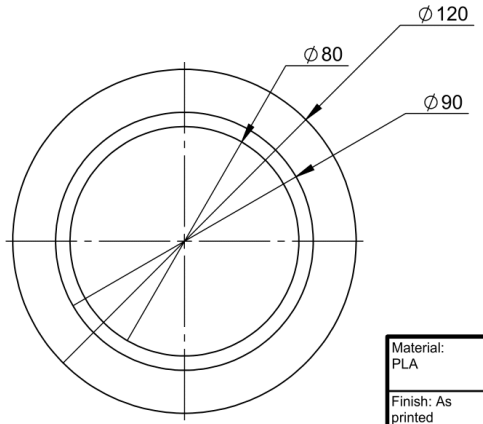


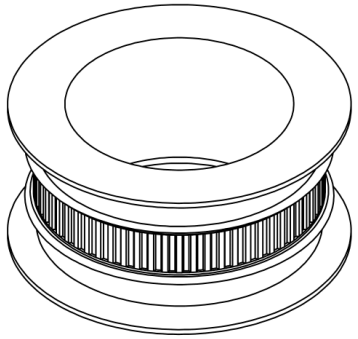
Material: PLA			Description: Servo elbow bracket for 3D printing
Finish: As printed			
Unless Otherwise Stated: Tol.: **±0.1** All Dimensions: **mm**		Drawn by: Nathan	Drawn Date: 10/05/25
	Document Type:	Checked/Approved by:	Checked/Approved Date:
	Masters Project		
Approx Weight: 0.039Kg		Part Number:	
Projection Method: **THIRD ANGLE**	Sheet Size: **A4**	Drawing Number:	Sheet: **1 of 1** / Revision:





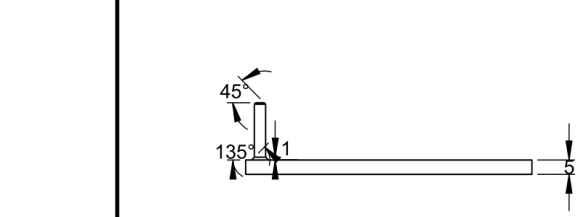


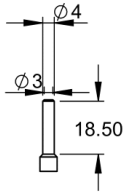


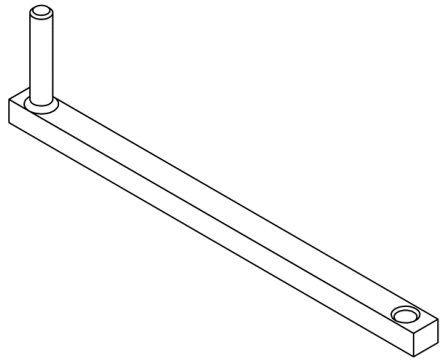


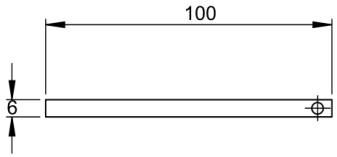
Material: PLA		Description: wrist collar for 3D printing
Finish: As printed		
Unless Otherwise Stated: Tol.: ±0.1 All Dimensions: mm	Document Type:	Drawn by: Nathan / Drawn Date:10/05/25
	Masters Project	Checked/Approved by: / Checked/Approved Date:
Approx Weight: 0.219Kg		Part Number:
Projection Method: THIRD ANGLE / Sheet Size: A4		Drawing Number: / Sheet: 1 of 1 / Revision:

SOLIDWORKS Educational Product. For Instructional Use Only.

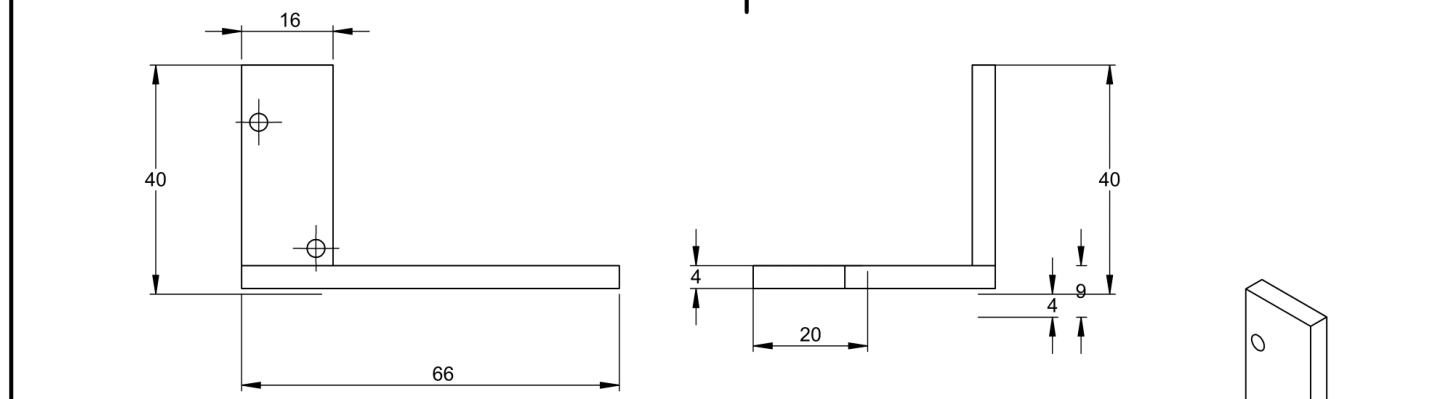




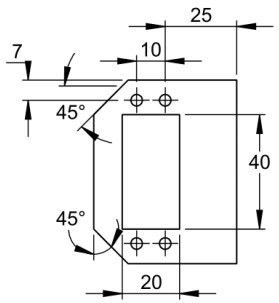


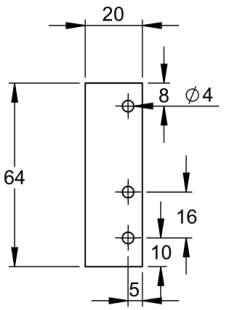


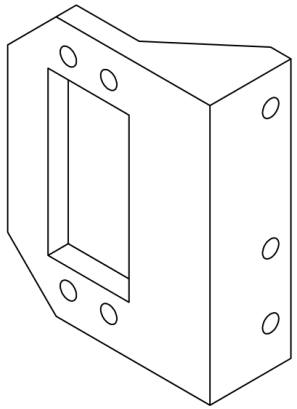
Material: PLA		Description: Wrist roation bar for 3D printing
Finish: As printed		
Unless Otherwise Stated: Tol.: ±0.1 All Dimensions: mm	Document Type:	Drawn by: Nathan / Drawn Date: 10/05/25
	Masters Project	Checked/Approved by: / Checked/Approved Date:
Approx Weight: 0.008Kg		Part Number:
Projection Method: THIRD ANGLE / Sheet Size: A4		Drawing Number: / Sheet: 1 of 1 / Revision:

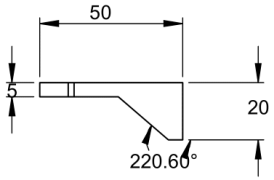


Material: PLA		Description: Limit switch mounting bracket for 3D printing
Finish: As printed		
Unless Otherwise Stated: Tol.: ±0.1 All Dimensions: mm	Document Type:	Drawn by: Nathan — Drawn Date: 10/05/25
	Masters Project	Checked/Approved by: — Checked/Approved Date:
Approx Weight: 0.015Kg		Part Number:
Projection Method: THIRD ANGLE	Sheet Size: A4	Drawing Number: — Sheet: 1 of 1 — Revision:

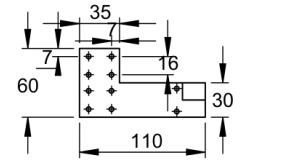


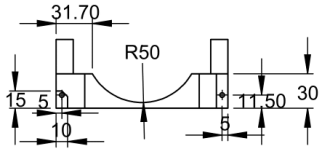


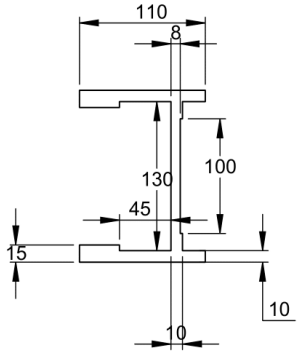


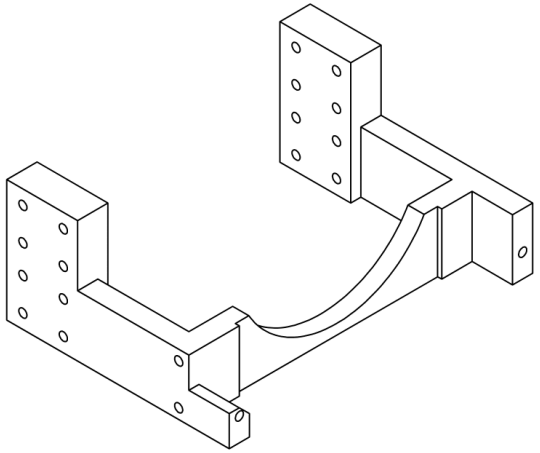


Material: PLA		Description: wrist servo mount for 3D printing
Finish: As printed		
Unless Otherwise Stated: Tol.: ±0.1 All Dimensions: mm	Document Type:	Drawn by: Nathan / Drawn Date:10/05/25
	Masters Project	Checked/Approved by: / Checked/Approved Date:
Approx Weight: 0.025Kg		Part Number:
Projection Method: THIRD ANGLE / Sheet Size: **A4**		Drawing Number: / Sheet: **1 of 1** / Revision:

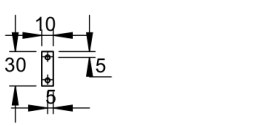


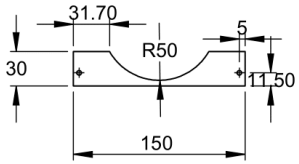


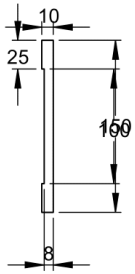


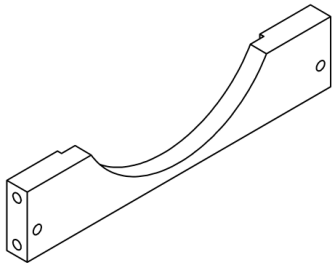


Material: PLA		Description: wrist mount for 3D printing
Finish: As printed		
Unless Otherwise Stated: Tol.: ±0.1 All Dimensions: mm	Document Type:	Drawn by: Nathan / Drawn Date: 10/05/25
	Masters Project	Checked/Approved by: / Checked/Approved Date:
Approx Weight: 0.240Kg		Part Number:
Projection Method: THIRD ANGLE	Sheet Size: A4	Drawing Number: / Sheet: 1 of 1 / Revision:

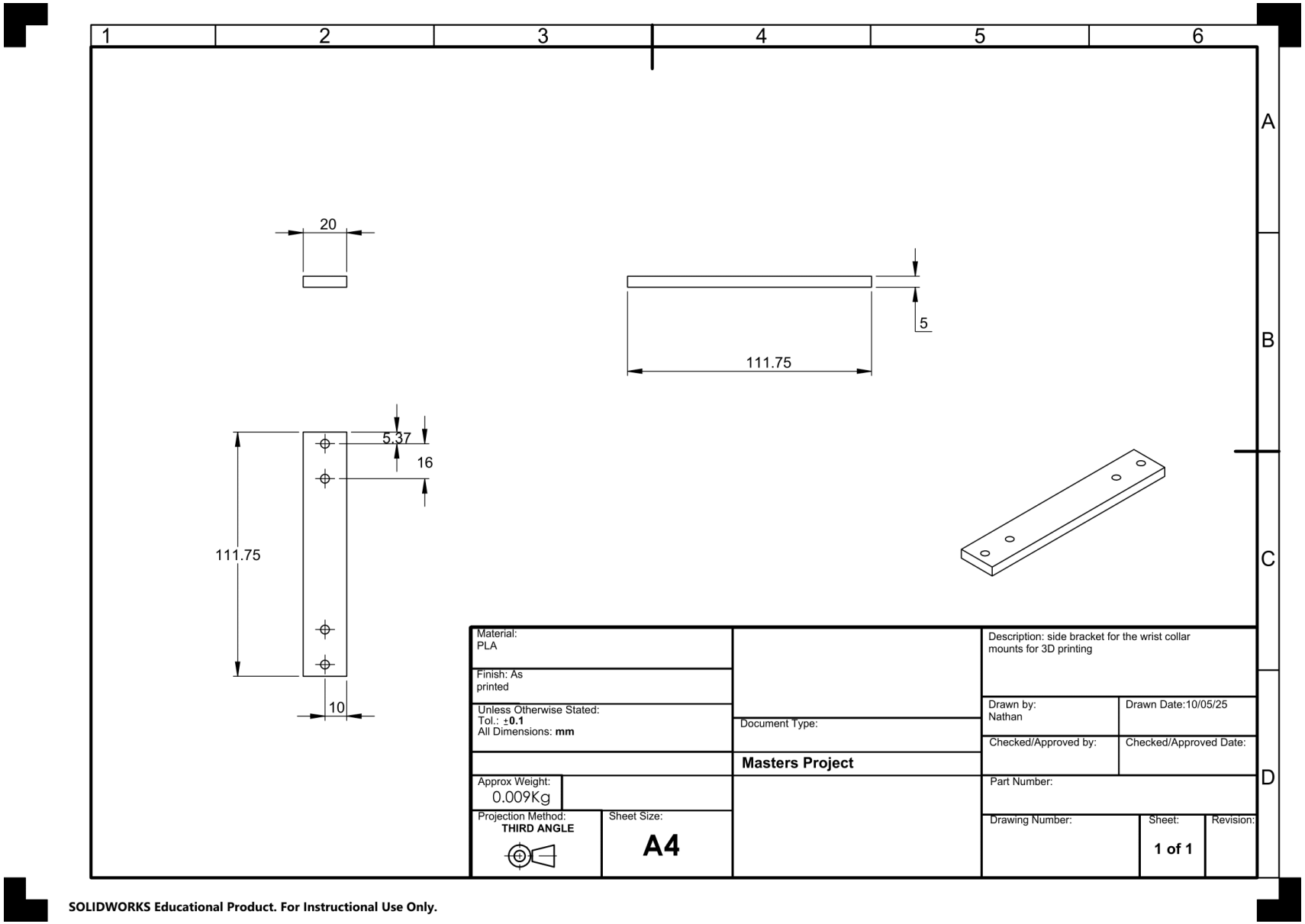








Material: PLA		Description: wrist collar top mount for 3D printing
Finish: As printed		
Unless Otherwise Stated: Tol.: ±0.1 All Dimensions: mm	Document Type:	Drawn by: Nathan / Drawn Date:10/05/25
	Masters Project	Checked/Approved by: / Checked/Approved Date:
Approx Weight: 0.027Kg		Part Number:
Projection Method: THIRD ANGLE / Sheet Size: A4		Drawing Number: / Sheet: 1 of 1 / Revision:



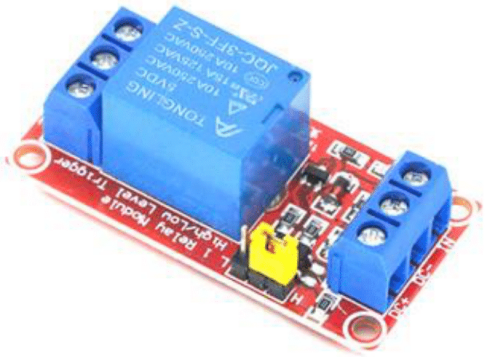
Appendix B – 5v Relay Board Datasheet

Handson Technology

User Guide

1-Channel 5V Optical Isolated Relay Module

This is a Lo/Hi Level 5V (user configurable) 1-channel relay interface board, and each channel needs a 15-20mA driver current. It can be used to control various appliances and equipment with large current. It is equipped with high-current relays that work under AC250V 10A or DC30V 10A. It has a standard interface that can be controlled directly by microcontroller. This module is optically isolated from high voltage side for safety requirement and also prevent ground loop when interface to microcontroller.



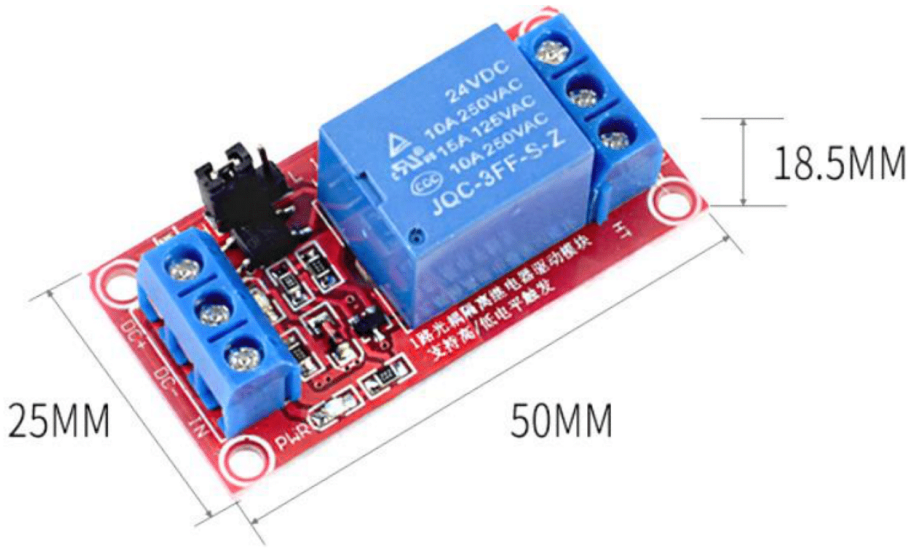
SKU: MDU1150

Brief Data:

- Relay Maximum output: DC 30V/10A, AC 250V/10A.
- Triggering Level: High & Low (Jumper Setting).
- 1 Channel Relay Module with Opto-coupler Isolation.
- On-board power and trigger LED Indictor.
- Standard interface that can be controlled directly by microcontroller (8051, AVR, *PIC, DSP, ARM, ARM, MSP430, TTL logic).
- Relay of high quality low noise relays SPDT. A common terminal, a normally open, one normally closed terminal.
- Opto-Coupler isolation, for high voltage safety and prevent ground loop with microcontroller.

Mechanical Dimension:

Unit: mm

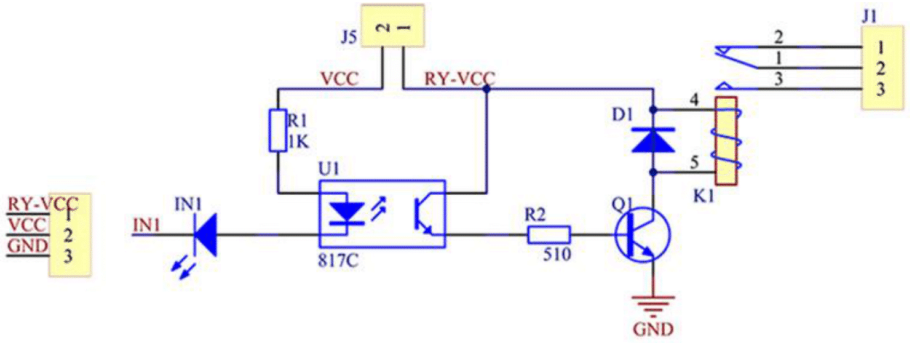


Schematic:

VCC and RY-VCC are also the power supply of the relay module. When you need to drive a large power load, you can take the jumper cap off and connect an extra power to RY-VCC to supply the relay; connect VCC to 5V of the MCU board to supply input signals.

NOTES: If you want complete optical isolation, connect "Vcc" to Arduino +5 volts but do NOT connect Arduino Ground. Remove the Vcc to JD-Vcc jumper. Connect a separate +5 supply to "JD-Vcc" and board Gnd. This will supply power to the transistor drivers and relay coils.

If relay isolation is enough for your application, connect Arduino +5 and Gnd, and leave Vcc to JD-Vcc jumper in place.



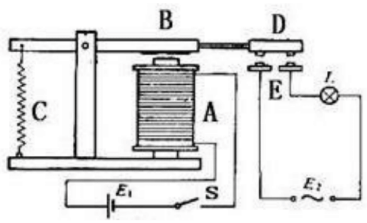
It is sometimes possible to use this relay boards with 3.3V signals, if the JD-VCC (Relay Power) is provided from a +5V supply and the VCC to JD-VCC jumper is removed. That 5V relay supply could be totally isolated from the 3.3V device, or have a common ground if opto-isolation is not needed. If used with isolated 3.3V signals, VCC (To the input of the opto-isolator, next to the IN pins) should be connected to the 3.3V device's +3.3V supply.

NOTE: Some Raspberry-Pi users have found that some relays are reliable and others do not actuate sometimes. It may be necessary to change the value of R1 from 1000 ohms to something like 220 ohms, or supply +5V to the VCC connection.

NOTE: The digital inputs from Arduino are Active LOW: The relay actuates and LED lights when the input pin is LOW, and turns off on HIGH.

Operating Principle:

See the picture below: A is an electromagnet, B armature, C spring, D moving contact, and E fixed contacts. There are two fixed contacts, a normally closed one and a normally open one. When the coil is not energized, the normally open contact is the one that is off, while the normally closed one is the other that is on.



Supply voltage to the coil and some currents will pass through the coil thus generating the electromagnetic effect. So the armature overcomes the tension of the spring and is attracted to the core, thus closing the moving contact of the armature and the normally open (NO) contact or you may say releasing the former and the normally closed (NC) contact. After the coil is de-energized, the electromagnetic force disappears and the armature moves back to the original position, releasing the moving contact and normally closed contact. The closing and releasing of the contacts results in power on and off of the circuit.

Input:

VCC : Connected to positive supply voltage (supply power according to relay voltage)

GND : Connected to supply ground.

IN1: Signal triggering terminal 1 of relay module

IN2: Signal triggering terminal 2 of relay module

Output:

Each module of the relay has one NC (normally close), one NO (normally open) and one COM (Common) terminal. So there are 2 NC, 2 NO and 2 COM of the channel relay in total. NC stands for the normal close port contact and the state without power. NO stands for the normal open port contact and the state with

power. COM means the common port. You can choose NC port or NO port according to whether power or not.

Testing Setup:

When a low level is supplied to signal terminal of the 2-channel relay, the LED at the output terminal will light up. Otherwise, it will turn off. If a periodic high and low level is supplied to the signal terminal, you can see the LED will cycle between on and off.

For Arduino：

Step 1：

Connect the signal terminal IN1、IN2 of 2-channel relay to digital pin 4 & 5 of the Arduino Uno or ATMega2560 board, and connect an LED at the output terminal.

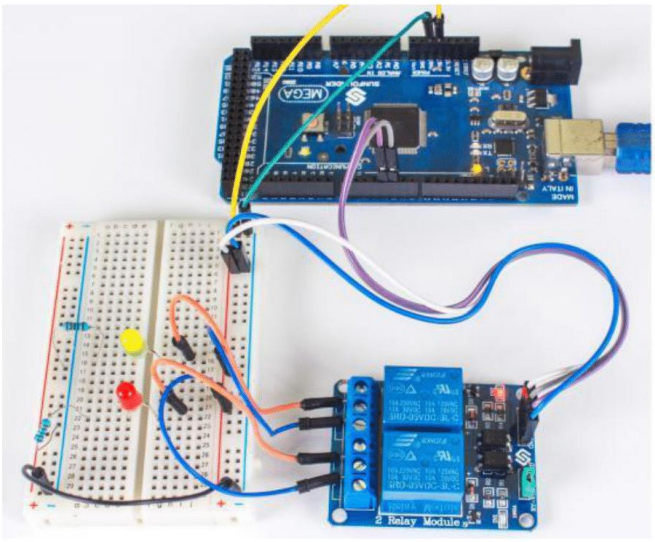
IN1> 4

IN2> 5

Step 2:

Upload the sketch "text_code" to the Arduino Uno or ATMega2560 board. Then you can see the LED cycle between on and off.

The actual figure is shown below:



For raspberry Pi:

Step1:

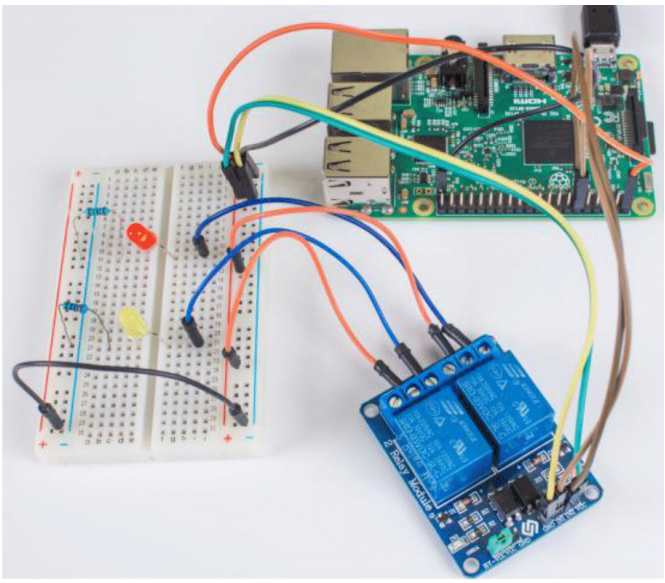
Connect the signal terminal IN2、IN1 of 2-channel relay to port 17、18 of the Raspberry Pi, and connect an LED at the output terminal.

IN2 > 17

IN1 > 18

Step 2:

Run the “test_code”. Then you can see the LED cycle between on and off.



Sketch for Arduino:

```
/*************************************************
   Name: _2_channel_relay
   Description: control the 2 channel relay module to ON or OFF
   Website: www.handsontec.com
   Email: techsupport@handsontec.com
**************************************************/

//the relays connect to
int IN1 = 4;
int IN2 = 5;

#define ON   0
#define OFF  1

void setup()
{
  relay_init();//initialize the relay
}

void loop()  {
  relay_SetStatus(ON, OFF);//turn on RELAY_1
  delay(2000);//delay 2s
  relay_SetStatus(OFF, ON);//turn on RELAY_2
  delay(2000);//delay 2s
}
void relay_init(void)//initialize the relay
{
  //set all the relays OUTPUT
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  relay_SetStatus(OFF, OFF); //turn off all the relay
```

```
}
//set the status of relays
void relay_SetStatus( unsigned char status_1,  unsigned char status_2)
{
  digitalWrite(IN1, status_1);
  digitalWrite(IN2, status_2);
}
```

Code for Raspberry Pi:

```
#!/usr/bin/env python
'''
**********************************************************************
* Filename      : 2_channel_relay.py
* Description   : a sample script for 2-Channel High trigger Relay
* E-mail        : techsupport@handsontec.com
* Website       : www.handsontec.com
* Detail        : New file
**********************************************************************
'''
import RPi.GPIO as GPIO
from time import sleep

Relay_channel = [17, 18]

def setup():
    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(Relay_channel, GPIO.OUT, initial=GPIO.LOW)
    print "|=====================================================|"
    print "|          2-Channel High trigger Relay Sample        |"
    print "|-----------------------------------------------------|"
    print "|                                                     |"
    print "|           Turn 2 channels on off in orders          |"
    print "|                                                     |"
    print "|                      17 ===> IN2                    |"
    print "|                      18 ===> IN1                    |"
    print "|                                                     |"
    print "|                                                     |"
    print "|=====================================================|"

def main():
    while True:
        for i in range(0, len(Relay_channel)):
            print '...Relay channel %d on' % i+1
            GPIO.output(Relay_channel[i], GPIO.HIGH)
            sleep(0.5)
            print '...Relay channel %d off' % i+1
            GPIO.output(Relay_channel[i], GPIO.LOW)
            sleep(0.5)

def destroy():
    GPIO.output(Relay_channel, GPIO.LOW)
    GPIO.cleanup()

if __name__ == '__main__':
    setup()
    try:
        main()
    except KeyboardInterrupt:
        destroy()
```

Related Information:

- 2-Channel Solid State Relay (SSR) Module 2A-240VAC
- 30A High Power Optical Isolated Relay Module
- 4-Channel 5V Optical Isolated Relay Module
- 8 Channel 5V Optical Isolated Relay Module
- Photosensitive Light Activate Relay Module

HandsOn Technology provides a multimedia and interactive platform for everyone interested in electronics. From beginner to diehard, from student to lecturer. Information, education, inspiration and entertainment. Analog and digital, practical and theoretical; software and hardware.

HandsOn Technology support Open Source Hardware (OSHW) Development Platform.

Learn : Design : Share

www.handsontec.com

The Face behind our product quality…

In a world of constant change and continuous technological development, a new or replacement product is never far away – and they all need to be tested.
Many vendors simply import and sell wihtout checks and this cannot be the ultimate interests of anyone, particularly the customer. Every part sell on Handsotec is fully tested. So when buying from Handsontec products range, you can be confident you're getting outstanding quality and value.

We keep adding the new parts so that you can get rolling on your next project.

Breakout Boards & Modules

Connectors

Electro-Mechanical Parts

Engineering Material

Mechanical Hardware

Electronics Components

P

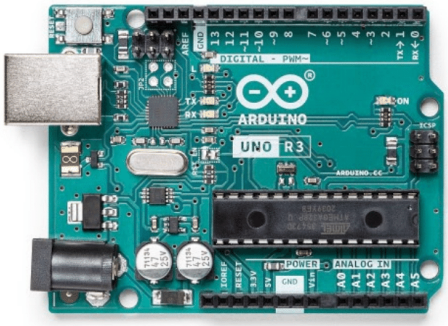
Power Supply

Arduino Board & Shield

Tools & Accessory

Appendix C – Arduino Uno R3

User Manual
SKU: A000066



Description

The Arduino® UNO R3 is the perfect board to get familiar with electronics and coding. This versatile development board is equipped with the well-known ATmega328P and the ATMega 16U2 Processor.

This board will give you a great first experience within the world of Arduino.

Target areas:

Maker, introduction, industries

Features

- **ATMega328P** Processor
  - **Memory**
    - AVR CPU at up to 16 MHz
    - 32 kB Flash
    - 2 kB SRAM
    - 1 kB EEPROM
  - **Security**
    - Power On Reset (POR)
    - Brown Out Detection (BOD)
  - **Peripherals**
    - 2x 8-bit Timer/Counter with a dedicated period register and compare channels
    - 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
    - 1x USART with fractional baud rate generator and start-of-frame detection
    - 1x controller/peripheral Serial Peripheral Interface (SPI)
    - 1x Dual mode controller/peripheral I2C
    - 1x Analog Comparator (AC) with a scalable reference input
    - Watchdog Timer with separate on-chip oscillator
    - Six PWM channels
    - Interrupt and wake-up on pin change
  - **ATMega16U2 Processor**
    - 8-bit AVR® RISC-based microcontroller
  - **Memory**
    - 16 kB ISP Flash
    - 512B EEPROM
    - 512B SRAM
    - debugWIRE interface for on-chip debugging and programming
  - **Power**
    - 2.7-5.5 volts

CONTENTS

1 The Board

1.1 Application Examples

The UNO board is the flagship product of Arduino. Regardless if you are new to the world of electronics or will use the UNO R3 as a tool for education purposes or industry-related tasks, the UNO R3 is likely to meet your needs.

First entry to electronics: If this is your first project within coding and electronics, get started with our most used and documented board; UNO. It is equipped with the well-known ATmega328P processor, 14 digital input/output pins, 6 analog inputs, USB connections, ICSP header and reset button. This board includes everything you will need for a great first experience with Arduino.

Industry-standard development board: Using the UNO R3 board in industries, there are a range of companies using the UNO R3 board as the brain for their PLC's.

Education purposes: Although the UNO R3 board has been with us for about ten years, it is still widely used for various education purposes and scientific projects. The board's high standard and top quality performance makes it a great resource to capture real time from sensors and to trigger complex laboratory equipment to mention a few examples.

1.2 Related Products

- Arduino Starter Kit
- Arduino UNO R4 Minima
- Arduino UNO R4 WiFi
- Tinkerkit Braccio Robot

2 Ratings

2.1 Recommended Operating Conditions

Symbol	Description	Min	Max
	Conservative thermal limits for the whole board:	-40 °C (-40 °F)	85 °C (185 °F)

NOTE: In extreme temperatures, EEPROM, voltage regulator, and the crystal oscillator, might not work as expected.

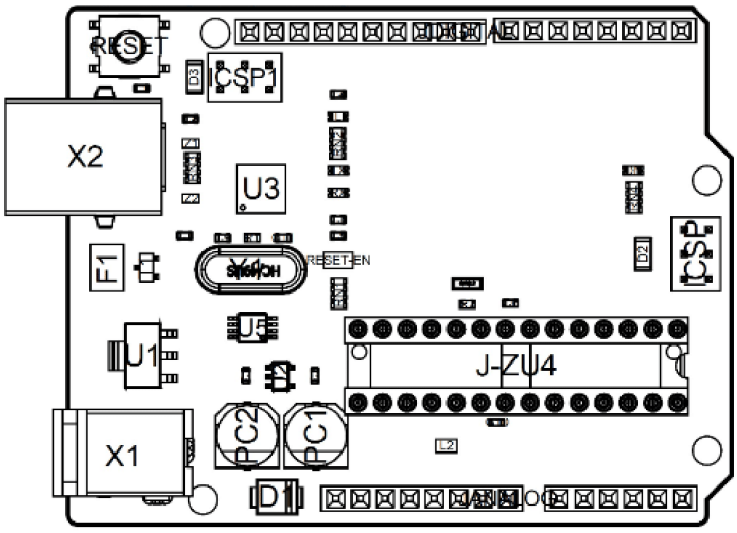
2.2 Power Consumption

Symbol	Description	Min	Typ	Max	Unit
VINMax	Maximum input voltage from VIN pad	6	-	20	V
VUSBMax	Maximum input voltage from USB connector		-	5.5	V
PMax	Maximum Power Consumption	-	-	xx	mA

3 Functional Overview

3.1 Board Topology

Top view



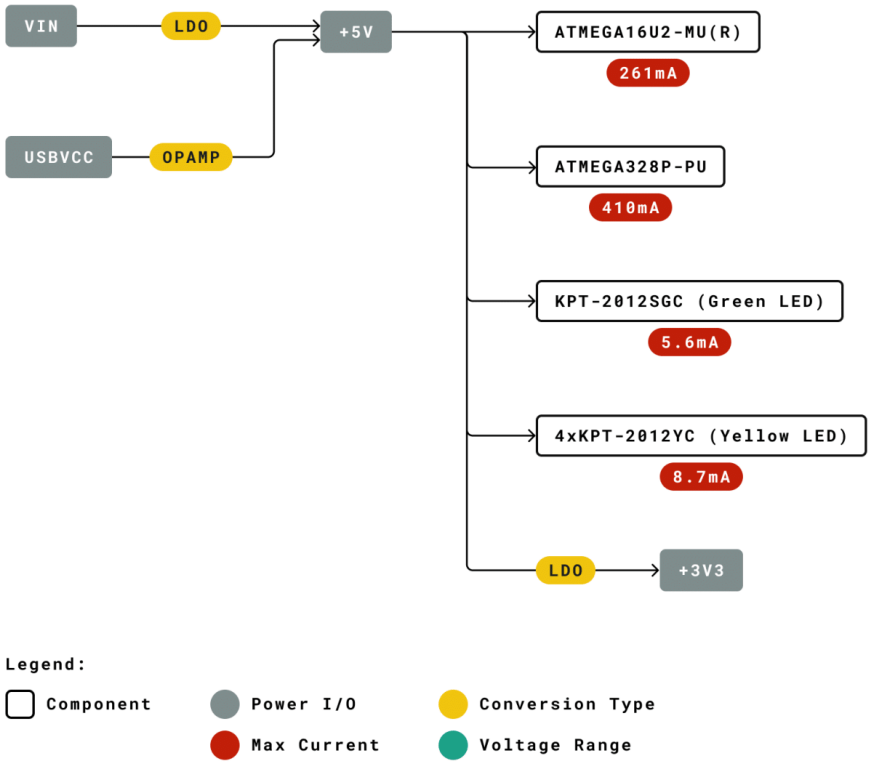
Board topology

Ref.	Description	Ref.	Description
X1	Power jack 2.1x5.5mm	U1	SPX1117M3-L-5 Regulator
X2	USB B Connector	U3	ATMEGA16U2 Module
PC1	EEE-1EA470WP 25V SMD Capacitor	U5	LMV358LIST-A.9 IC
PC2	EEE-1EA470WP 25V SMD Capacitor	F1	Chip Capacitor, High Density
D1	CGRA4007-G Rectifier	ICSP	Pin header connector (through hole 6)
J-ZU4	ATMEGA328P Module	ICSP1	Pin header connector (through hole 6)
Y1	ECS-160-20-4X-DU Oscillator		

3.2 Processor

The Main Processor is a ATmega328P running at up to 20 MHz. Most of its pins are connected to the external headers, however some are reserved for internal communication with the USB Bridge coprocessor.

3.3 Power Tree



Power tree

4 Board Operation

4.1 Getting Started - IDE

If you want to program your UNO R3 while offline you need to install the Arduino Desktop IDE [1] To connect the UNO R3 to your computer, you'll need a USB-B cable. This also provides power to the board, as indicated by the LED.

4.2 Getting Started - Arduino Cloud Editor

All Arduino boards, including this one, work out-of-the-box on the Arduino Cloud Editor [2], by just installing a simple plugin.

The Arduino Cloud Editor is hosted online, therefore it will always be up-to-date with the latest features and support for all boards. Follow **[3]** to start coding on the browser and upload your sketches onto your board.

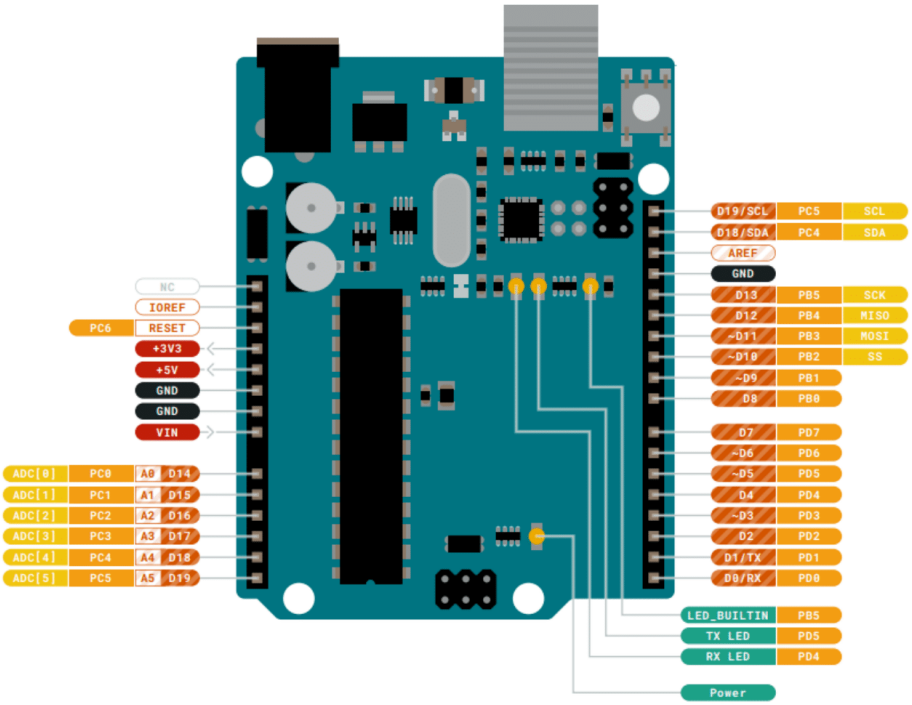
4.3 Sample Sketches

Sample sketches for the UNO R3 can be found either in the "Examples" menu in the Arduino IDE or in the "Documentation" section of the Arduino website [4].

4.4 Online Resources

Now that you have gone through the basics of what you can do with the board you can explore the endless possibilities it provides by checking exciting projects on Arduino Project Hub [5], the Arduino Library Reference [6] and the online Arduino store [7] where you will be able to complement your board with sensors, actuators and more.

5 Connector Pinouts



Pinout

5.1 JANALOG

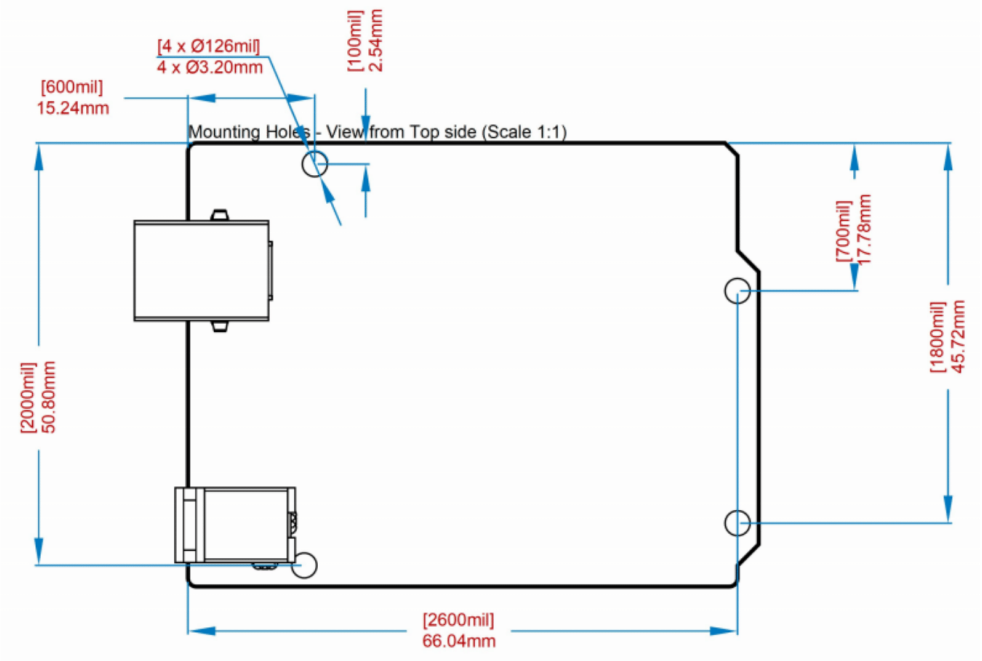
Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	Analog input 1 /GPIO
11	A2	Analog/GPIO	Analog input 2 /GPIO
12	A3	Analog/GPIO	Analog input 3 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

5.2 JDIGITAL

Pin	Function	Type	Description
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select
12	MOSI	Digital	SPI1 Main Out Secondary In
13	MISO	Digital	SPI Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Power	Ground
16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)

5.3 Mechanical Information

5.4 Board Outline & Mounting Holes



Board outline

6 Certifications

6.1 Declaration of Conformity CE DoC (EU)

We declare under our sole responsibility that the products above are in conformity with the essential requirements of the following EU Directives and therefore qualify for free movement within markets comprising the European Union (EU) and European Economic Area (EEA).

ROHS 2 Directive 2011/65/EU	
Conforms to:	EN50581:2012
Directive 2014/35/EU. (LVD)	
Conforms to:	EN 60950-1:2006/A11:2009/A1:2010/A12:2011/AC:2011
Directive 2004/40/EC & 2008/46/EC & 2013/35/EU, EMF	
Conforms to:	EN 62311:2008

6.2 Declaration of Conformity to EU RoHS & REACH 211 01/19/2021

Arduino boards are in compliance with RoHS 2 Directive 2011/65/EU of the European Parliament and RoHS 3 Directive 2015/863/EU of the Council of 4 June 2015 on the restriction of the use of certain hazardous substances in electrical and electronic equipment.

Substance	Maximum limit (ppm)
Lead (Pb)	1000
Cadmium (Cd)	100
Mercury (Hg)	1000
Hexavalent Chromium (Cr6+)	1000
Poly Brominated Biphenyls (PBB)	1000
Poly Brominated Diphenyl ethers (PBDE)	1000
Bis(2-Ethylhexyl} phthalate (DEHP)	1000
Benzyl butyl phthalate (BBP)	1000
Dibutyl phthalate (DBP)	1000
Diisobutyl phthalate (DIBP)	1000

Exemptions: No exemptions are claimed.

Arduino Boards are fully compliant with the related requirements of European Union Regulation (EC) 1907 /2006 concerning the Registration, Evaluation, Authorization and Restriction of Chemicals (REACH). We declare none of the SVHCs (https://echa.europa.eu/web/guest/candidate-list-table), the Candidate List of Substances of Very High Concern for authorization currently released by ECHA, is present in all products (and also package) in quantities totaling in a concentration equal or above 0.1%. To the best of our knowledge, we also declare that our products do not contain any of the substances listed on the "Authorization List" (Annex XIV of the REACH regulations) and Substances of Very High Concern (SVHC) in any significant amounts as specified by the Annex XVII of Candidate list published by ECHA (European Chemical Agency) 1907 /2006/EC.

6.3 Conflict Minerals Declaration

As a global supplier of electronic and electrical components, Arduino is aware of our obligations with regards to laws and regulations regarding Conflict Minerals, specifically the Dodd-Frank Wall Street Reform and Consumer Protection Act, Section 1502. Arduino does not directly source or process conflict minerals such as Tin, Tantalum, Tungsten, or Gold. Conflict minerals are contained in our products in the form of solder, or as a component in metal alloys. As part of our reasonable due diligence Arduino has contacted component suppliers within our supply chain to verify their continued compliance with the regulations. Based on the information received thus far we declare that our products contain Conflict Minerals sourced from conflict-free areas.

7 FCC Caution

Any Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

(1) This device may not cause harmful interference

(2) this device must accept any interference received, including interference that may cause undesired operation.

FCC RF Radiation Exposure Statement:

1. This Transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.
2. This equipment complies with RF radiation exposure limits set forth for an uncontrolled environment.
3. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body.

English: User manuals for license-exempt radio apparatus shall contain the following or equivalent notice in a conspicuous location in the user manual or alternatively on the device or both. This device complies with Industry Canada license-exempt RSS standard(s). Operation is subject to the following two conditions:

(1) this device may not cause interference

(2) this device must accept any interference, including interference that may cause undesired operation of the device.

French: Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes :

(1) l' appareil nedoit pas produire de brouillage

(2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

IC SAR Warning:

English This equipment should be installed and operated with minimum distance 20 cm between the radiator and your body.

French: Lors de l' installation et de l' exploitation de ce dispositif, la distance entre le radiateur et le corps est d 'au moins 20 cm.

Important: The operating temperature of the EUT can't exceed 85°C and shouldn't be lower than -40°C.

Hereby, Arduino S.r.l. declares that this product is in compliance with essential requirements and other relevant provisions of Directive 2014/53/EU. This product is allowed to be used in all EU member states.

8 Company Information

Company name	Arduino S.r.l
Company Address	Via Andrea Appiani 25 20900 MONZA Italy

9 Reference Documentation

Reference	Link
Arduino IDE (Desktop)	https://www.arduino.cc/en/Main/Software
Arduino Cloud Editor	https://create.arduino.cc/editor
Arduino Cloud Editor - Getting Started	https://docs.arduino.cc/arduino-cloud/guides/editor/
Arduino Website	https://www.arduino.cc/
Arduino Project Hub	https://create.arduino.cc/projecthub?by=part&part_id=11332&sort=trending
Library Reference	https://www.arduino.cc/reference/en/
Arduino Store	https://store.arduino.cc/

10 Revision History

Date	Revision	Changes
25/04/2024	3	Updated link to new Cloud Editor
26/07/2023	2	General Update
06/2021	1	Datasheet release

中文（ZH）

描述

Arduino UNO R3 是熟悉电子技术和编码的完美开发板。这款多功能开发板配备了著名的 ATmega328P 和 ATMega 16U2 处理器。该开发板将为您带来 Arduino 世界绝佳的初次体验。

目标领域：

创客、介绍、工业领域

特点

- **ATMega328P** 处理器
  - **内存**
    - AVR CPU 频率高达 16 MHz
    - 32KB 闪存
    - 2KB SRAM
    - 1KB EEPROM
  - **安全性**
    - 上电复位 (POR)
    - 欠压检测 (BOD)
  - **外设**
    - 2x 8 位定时器/计数器，带专用周期寄存器和比较通道
    - 1x 16 位定时器/计数器，带专用周期寄存器、输入捕获和比较通道
    - 1x USART，带分数波特率发生器和起始帧信号检测功能
    - 1x 控制器/外设串行外设接口 (SPI)
    - 1x 双模控制器/外设 I2C
    - 1 个模拟比较器 (AC)，带可扩展参考输入
    - 看门狗定时器，带独立的片上振荡器
    - 6 通道 PWM
    - 引脚变化时的中断和唤醒
  - **ATMega16U2 处理器**
    - 基于 AVR® RISC 的 8 位微控制器
  - **内存**
    - 16 KB ISP 闪存
    - 512B EEPROM
    - 512B SRAM

- 用于片上调试和编程的 debugWIRE 接口

- **电源**
  - 2.7-5.5 伏特

目录

11 电路板简介

11.1 应用示例

UNO 电路板是 Arduino 的旗舰产品。无论您是初次接触电路板产品，还是将 UNO 用作教育或工业相关任务的工具，UNO 都能满足您的需求。

初次接触电子技术: 如果这是您第一次参与编码和电子技术项目，那么就从我们最常用、记录最多的电路板 Arduino UNO 开始吧。它配备了著名的 ATmega328P 处理器、14 个数字输入/输出引脚、6 个模拟输入、USB 连接、ICSP 接头和复位按钮。该电路板包含了您获得良好的 Arduino 初次体验所需的一切。

** 行业标准开发板:** 在工业领域使用 Arduino UNO R3 开发板，有许多公司使用 UNO 开发板作为其 PLC 的大脑。

教育用途: 尽管我们推出 UNO R3 电路板已有大约十年之久，但它仍被广泛用于各种教育用途和科学项目。该电路板的高标准和一流性能使其成为从传感器采集实时数据和触发复杂实验室设备等各种应用场合的绝佳资源。

11.2 相关产品

- Starter Kit
- Arduino UNO R4 Minima
- Arduino UNO R4 WiFi
- Tinkerkit Braccio Robot

12 额定值

12.1 建议运行条件

符号	描述	最小值	最大值
	整个电路板的保守温度极限值：	-40 °C (-40°F)	85 °C (185°F)

注意： 在极端温度下，EEPROM、电压调节器和晶体振荡器可能无法正常工作。

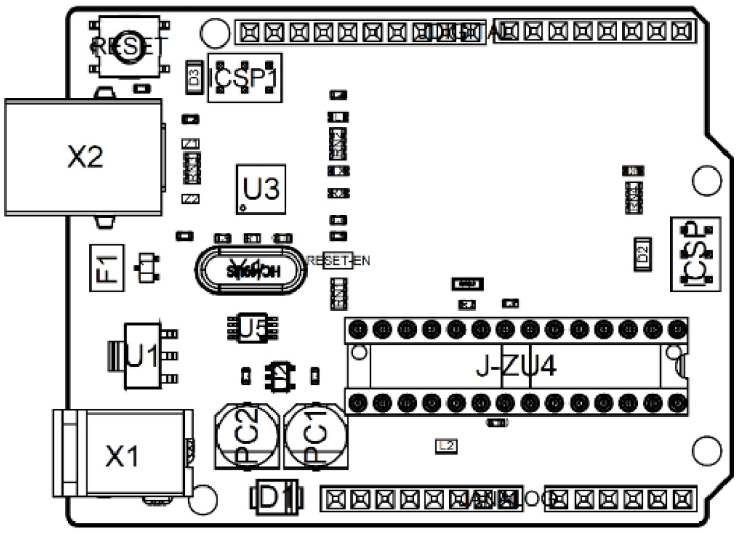
12.2 功耗

符号	描述	最小值	典型值	最大值	单位
VINMax	来自 VIN 焊盘的最大输入电压	6	-	20	V
VUSBMax	来自 USB 连接器的最大输入电压		-	5.5	V
PMax	最大功耗	-	-	xx	mA

13 功能概述

13.1 电路板拓扑结构

俯视图



电路板拓扑结构

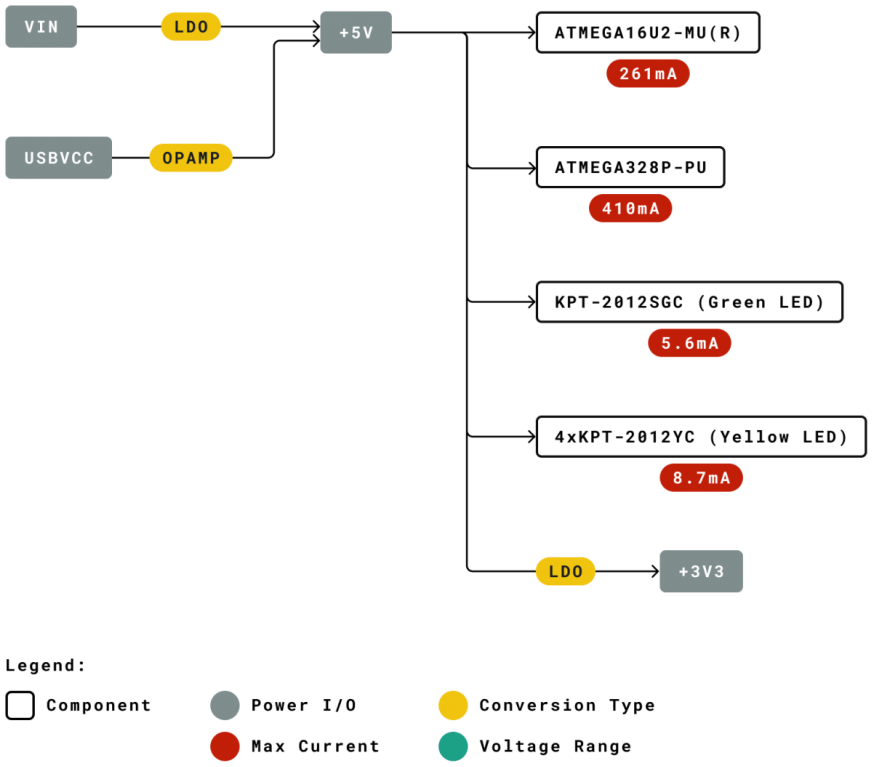
编号	描述	编号	描述
X1	电源插孔 2.1x5.5 毫米	U1	SPX1117M3-L-5 调节器

编号	描述	编号	描述
X2	USB B 连接器	U3	ATMEGA16U2 模块
PC1	EEE-1EA470WP 25V SMD 电容器	U5	LMV358LIST-A.9 IC
PC2	EEE-1EA470WP 25V SMD 电容器	F1	片式电容器，高密度
D1	CGRA4007-G 整流器	ICSP	引脚接头连接器（通过 6 号孔）
J-ZU4	ATMEGA328P 模块	ICSP1	引脚接头连接器（通过 6 号孔）
Y1	ECS-160-20-4X-DU 振荡器		

13.2 处理器

主处理器是 ATmega328P，运行频率高达 20 MHz。它的大部分引脚都与外部接头相连，但也有一些引脚用于与 USB 桥协处理器进行内部通信。

13.3 电源树



电源树

14 电路板操作

14.1 入门指南 - IDE

如需在离线状态下对 Arduino UNO R3 进行编程，则需要安装 Arduino Desktop IDE [1] 若要将 Arduino UNO 连接到计算机，需要使用 USB-B 电缆。如 LED 指示灯所示，该电缆还可以为电路板供电。

14.2 入门指南 - Arduino Cloud Editor

包括本电路板在内的所有 Arduino 电路板，都可以在 Arduino Cloud Editor [2] 上开箱即用，只需安装一个简单的插件即可。

Arduino Cloud Editor 是在线托管的，因此它将始终提供最新功能并支持所有电路板。接下来**[3]**开始在浏览器上编码并将程序上传到您的电路板上。

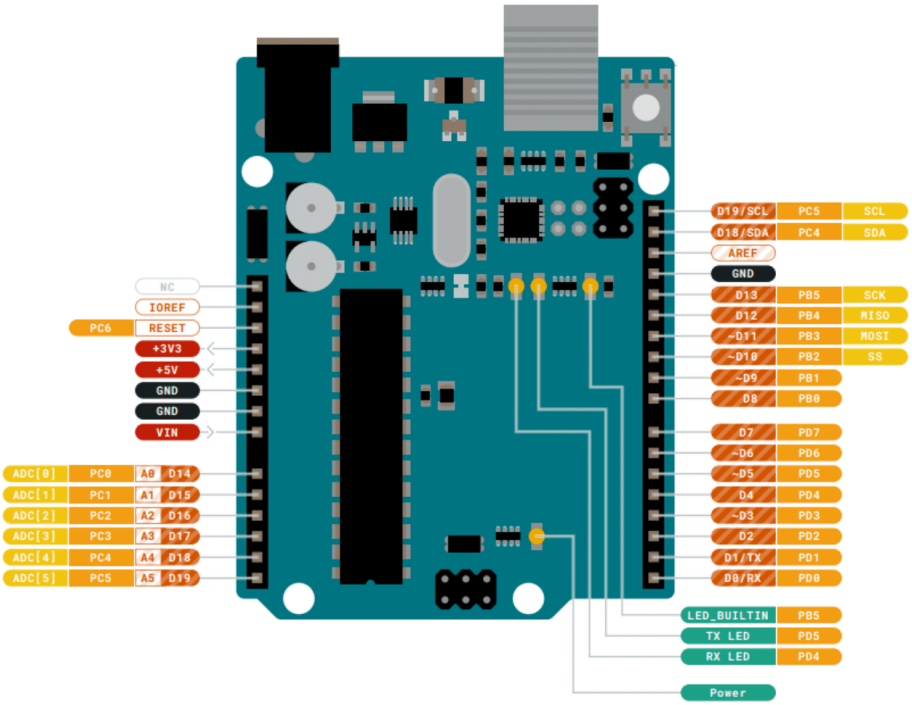
14.3 示例程序

Arduino UNO R3 的示例程序可以在 Arduino IDE 的“示例”菜单或 Arduino 网站 [4] 的“文档”部分找到

14.4 在线资源

现在，您已经了解该电路板的基本功能，就可以通过查看 Arduino Project Hub **[5]**、Arduino Library Reference **[6]** 以及在线 Arduino 商店 **[7]**上的精彩项目来探索它所提供的无限可能性；在这些项目中，您可以为电路板配备传感器、执行器等。

15 连接器引脚布局



布局

15.1 JANALOG

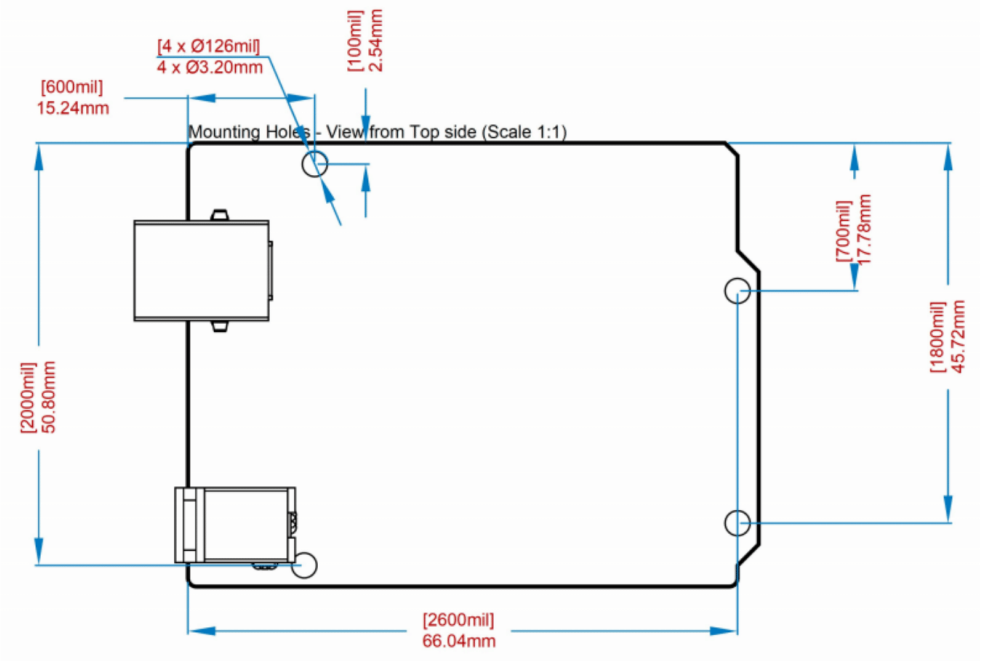
引脚	功能	类型	描述
1	NC	NC	未连接
2	IOREF	IOREF	数字逻辑参考电压 V - 连接至 5V
3	复位	复位	复位
4	+3V3	电源	+3V3 电源轨
5	+5V	电源	+5V 电源轨
6	GND	电源	接地
7	GND	电源	接地
8	VIN	电源	电压输入
9	A0	模拟/GPIO	模拟输入0 / GPIO
10	A1	模拟/GPIO	模拟输入1 / GPIO
11	A2	模拟/GPIO	模拟输入2 / GPIO
12	A3	模拟/GPIO	模拟输入3 / GPIO
13	A4/SDA	模拟输入/I2C	模拟输入 4/I2C 数据线
14	A5/SCL	模拟输入/I2C	模拟输入 5/I2C 时钟线

15.2 JDIGITAL

引脚	功能	类型	描述
1	D0	数字引脚/GPIO	数字引脚 0/GPIO
2	D1	数字引脚/GPIO	数字引脚 1/GPIO
3	D2	数字引脚/GPIO	数字引脚 2/GPIO
4	D3	数字引脚/GPIO	数字引脚 3/GPIO
5	D4	数字引脚/GPIO	数字引脚 4/GPIO
6	D5	数字引脚/GPIO	数字引脚 5/GPIO
7	D6	数字引脚/GPIO	数字引脚 6/GPIO
8	D7	数字引脚/GPIO	数字引脚 7/GPIO
9	D8	数字引脚/GPIO	数字引脚 8/GPIO
10	D9	数字引脚/GPIO	数字引脚 9/GPIO
11	SS	数字	SPI 芯片选择
12	MOSI	数字	SPI1 主输出副输入
13	MISO	数字	SPI 主输入副输出
14	SCK	数字	SPI 串行时钟输出
15	GND	电源	接地
16	AREF	数字	模拟参考电压
17	A4/SD4	数字	模拟输入 4/I2C 数据线（重复）
18	A5/SD5	数字	模拟输入 5/I2C 时钟线（重复）

15.3 机械层信息

15.4 电路板外形图和安装孔



电路板外形图

16 认证

16.1 符合性声明 CE DoC（欧盟）

我们在此郑重声明，上述产品符合以下欧盟指令的基本要求，因此有资格在包括欧盟（EU）和欧洲经济区（EEA）在内的市场内自由流通。

RoHS 2 指令 2011/65/EU	
符合：	EN50581:2012
指令 2014/35/EU。 (LVD)	
符合：	EN 60950-1:2006/A11:2009/A1:2010/A12:2011/AC:2011
指令 2004/40/EC & 2008/46/EC & 2013/35/EU, EMF	
符合：	EN 62311:2008

16.2 声明符合欧盟 RoHS 和 REACH 211 01/19/2021

Arduino 电路板符合欧洲议会关于限制在电子电气设备中使用某些有害物质的 RoHS 2 指令 2011/65/EU 和欧盟理事会于 2015 年 6 月 4 日颁布的关于限制在电子电气设备中使用某些有害物质的 RoHS 3 指令 2015/863/EU。

物质	最大限值 (ppm)
铅 (Pb)	1000
镉 (Cd)	100
汞 (Hg)	1000
六价铬（Cr6+）	1000
多溴联苯（PBB）	1000
多溴联苯醚（PBDE）	1000
邻苯二甲酸二(2-乙基己)酯 (DEHP)	1000
邻苯二甲酸丁苄酯 (BBP)	1000
邻苯二甲酸二丁酯（DBP）	1000
邻苯二甲酸二异丁酯（DIBP）	1000

豁免：未申请任何豁免。

Arduino 电路板完全符合欧盟法规 (EC) 1907/2006 中关于化学品注册、评估、许可和限制 (REACH) 的相关要求。我们声明，所有产品（包括包装）中的 SVHC (https://echa.europa.eu/web/guest/candidate-list-table)，（欧洲化学品管理局目前发布的《高度关注物质候选授权清单》）含量总浓度均未超过 0.1%。据我们所知，我们还声明，我们的产品不含 ECHA（欧洲化学品管理局）1907/2006/EC 公布的候选清单附件 XVII 中规定的“授权清单”（REACH 法规附件 XIV）和高度关注物质 (SVHC) 所列的任何物质。

16.3 冲突矿产声明

作为电子和电气元件的全球供应商，Arduino 意识到我们有义务遵守有关冲突矿产的法律法规，特别是《多德-弗兰克华尔街改革与消费者保护法案》第 1502 条。Arduino 不直接采购或加工锡、钽、钨或金等冲突矿物。冲突矿物以焊料的形式或作为金属合金的组成部分存在于我们的产品中。作为我们合理尽职调查的一部分，Arduino 已联系供应链中的元件供应商，以核实他们是否始终遵守法规的相关规定。根据迄今收到的信息，我们声明我们的产品中含有来自非冲突地区的冲突矿物。

17 FCC 警告

任何未经合规性负责方明确批准的更改或修改都可能导致用户无权操作设备。

本设备符合 FCC 规则第 15 部分的规定。操作须满足以下两个条件：

(1) 此设备不会造成有害干扰

(2) 此设备必须接受接收到的任何干扰，包括可能导致不良操作的干扰。

FCC 射频辐射暴露声明:

1. 此发射器不得与任何其他天线或发射器放置在同一位置或同时运行。
2. 此设备符合为非受控环境规定的射频辐射暴露限值。
3. 安装和操作本设备时，辐射源与您的身体之间至少应保持 20 厘米的距离。

English: User manuals for license-exempt radio apparatus shall contain the following or equivalent notice in a conspicuous location in the user manual or alternatively on the device or both. This device complies with Industry Canada license-exempt RSS standard(s). Operation is subject to the following two conditions:

(1) this device may not cause interference

(2) this device must accept any interference, including interference that may cause undesired operation of the device.

French: Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes :

(1) l' appareil nedoit pas produire de brouillage

(2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

IC SAR警告:

English This equipment should be installed and operated with a minimum distance of 20 cm between the radiator and your body.

French: Lors de l' installation et de l' exploitation de ce dispositif, la distance entre le radiateur et le corps est d 'au moins 20 cm.

重要提示： EUT 的工作温度不能超过 85°C，也不能低于 -40°C。

Arduino S.r.l. 特此声明，本产品符合 2014/53/EU 指令的基本要求和其他相关规定。本产品允许在所有欧盟成员国使用。

18 公司信息

公司名称	Arduino S.r.l
公司地址	Via Andrea Appiani 25 20900 MONZA Italy

19 参考资料

参考资料	链接
Arduino IDE (Desktop)	https://www.arduino.cc/en/Main/Software
Arduino IDE (Cloud)	https://create.arduino.cc/editor
Cloud IDE 入门指南	https://create.arduino.cc/projecthub/Arduino_Genuino/getting-started-with-arduino-web-editor-4b3e4a
Arduino 网站	https://www.arduino.cc/
Arduino Project Hub	https://create.arduino.cc/projecthub?by=part&part_id=11332&sort=trending
库参考	https://www.arduino.cc/reference/en/
在线商店	https://store.arduino.cc/

20 修订记录

日期	版次	变更
2023/07/26	2	一般更新
2021/06	1	数据表发布

Appendix D – DS3240 Servo Datasheet

DS3240 **DATASHEET 2024**

DS3240 Series Datasheet 2024

www.robotics.org.za

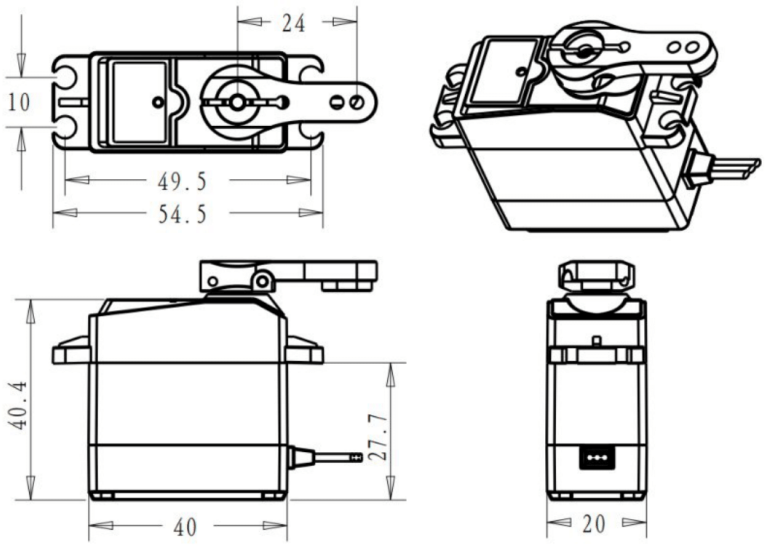
Product Name): DS3240 (Red)
Product Description): 6V 40kg RC Digital Servo

Model Numbers

DS3240 - Digital Servo 40kg.cm 180 Rotation Angle (3 Pin)
DS3240-270 - Digital Servo 40kg.cm 270 Rotation Angle (3 Pin)
DS3240-270-FB - Digital Servo 40kg.cm 270 Rotation Angle with Feedback (4 Pin)

DS3240 Series Datasheet 2024

Product Name)： DS3240（Red）
Product Description)： 6V 40kg RC Digital Servo



1. Apply Environmental Condition

No.	Item	Specification
1-1	存储温度 Storage Temperature Range	-30℃～80℃
1-2	运行温度 Operating Temperature Range	-25℃～70℃
1-3	工作电压范围 Operating Voltage Range	4.8-6.8V

2. 机械特性 Mechanical Specification

No.	Item	Specification
2-1	尺寸 Size	40*20*40.5mm
2-2	重量 Weight	60g
2-3	齿轮比 Gear ratio	342
2-4	轴承 Bearing	Double bearing
2 - 5	舵机线 Connector wire 300±5mm	
2-6	马达 Motor	3-pole(s)
2-7	防水性能 Waterproof performance	IP66

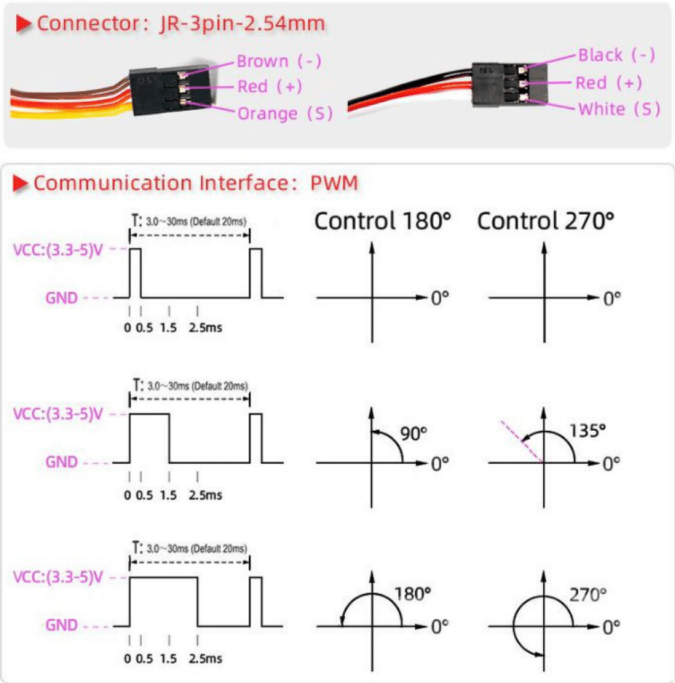
3. 电气特性 Electrical Specification No.

工作电压 Operating Voltage		5V	6.8V
3-1	待机电流 Idle current(at stopped)	4mA	5mA
3-2	空载转速 Operating speed (at no load)	0.20 sec/60°	0.17 sec/60°
3-3	堵转扭矩 Stall torque (at locked)	36 kg-cm	45 kg-cm
3-4	堵转电流 Stall current (at locked)	3.1A	3.9A

4. 控制特性 Control Specification

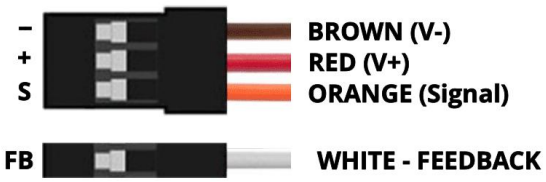
No.	Item	Specification
4-1	驱动方式 Control System	PWM(Pulse width modification)
4-2	脉宽范围 Pulse width range	500～2500μsec
4-3	中点位置 Neutral position	1500μsec
4-4	控制角度 Running degree	180° or 270° (when 500～2500 μ sec)
4-5	控制精度 Dead band width	3 μsec
4-6	控制频率 Operating frequency	50-330Hz
4-7	旋转方向 Rotating direction	Counterclockwise (when 500～2500 μsec)

5. 关于 PWM 控制说明 About PWM Control

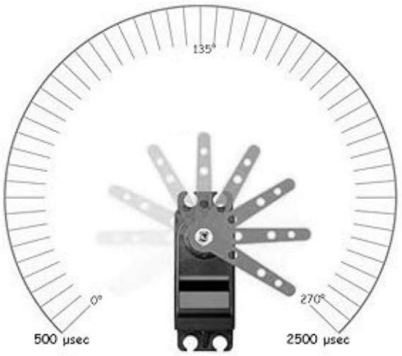


6. Color Coding

7. Feedback Wire



8. PWM Signal



Angle Control 270°

Angle Control 270 degree

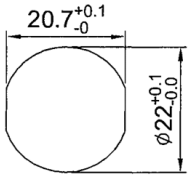
- Pulse width range: 500 ~ 2500µsec
- Rotating direction: Counterclockwise (when 500 ~ 2500 µsec)
- Neutral position: 1500µsec
- Running degree: 0° (when 500 µ sec)
- Running degree: 270° (when 2500 µ sec)

9. Internal Gear Structure

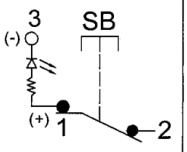
Appendix E – E-Stop Datasheet

Rating	21A 14V DC
Contact Resistance	50m ohm Max.
Insulation Resistance	DC 500V 100M ohm Min.
Dielectric Strength	AC 1500V 1 minute
Operation Temperature	-20°C~85°C
Switch Function	3P SPST-NC

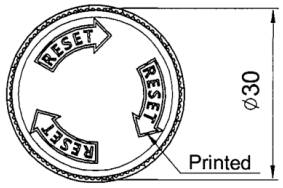
Mounting Hole

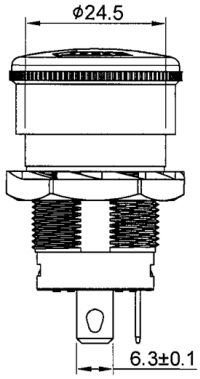


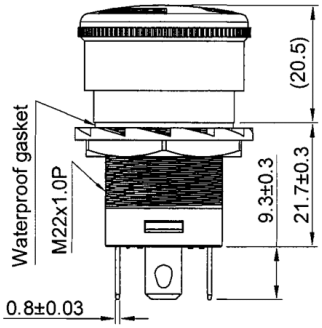
Circuit Diagram

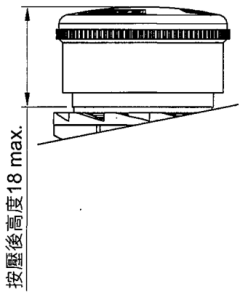


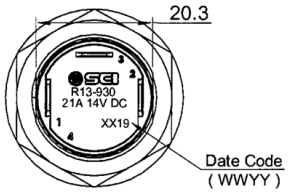
*Material:
1.Frame: Nylon-66 (94V-2).
2.Pushbutton: PC, transparent red.
3.Bottom: Nylon-66 (94V-2).
4.Terminal: Silver plated copper.
5.Waterproof gasket : Epdm.
*Sealed: IP56 when supplied with panel seal.











原稿
2019.9.05
研發部

Tolerance: ±0.3mm unless otherwise specificed.

Date	Name	Description				No.
Drawn	Checked	Approved	SCI BOM No.	1910103	Model Name	Emergency Stop Switch
Jam Chan			Customer No.		Model No.	R13-930LS-01-BRR-1 Rev. 0
11/22/'18			Unit	mm	Shin Chin Industrial Co., Ltd.	

Appendix F – INA219 Datasheet

Zerø-Drift, Bi-Directional CURRENT/POWER MONITOR with I²C™ Interface

Check for Samples: INA219

FEATURES

- SENSES BUS VOLTAGES FROM 0V TO +26V
- REPORTS CURRENT, VOLTAGE, AND POWER
- 16 PROGRAMMABLE ADDRESSES
- HIGH ACCURACY: 0.5% (Max) OVER TEMPERATURE (INA219B)
- FILTERING OPTIONS
- CALIBRATION REGISTERS
- SOT23-8 AND SO-8 PACKAGES

APPLICATIONS

- SERVERS
- TELECOM EQUIPMENT
- NOTEBOOK COMPUTERS
- POWER MANAGEMENT
- BATTERY CHARGERS
- WELDING EQUIPMENT
- POWER SUPPLIES
- TEST EQUIPMENT

DESCRIPTION

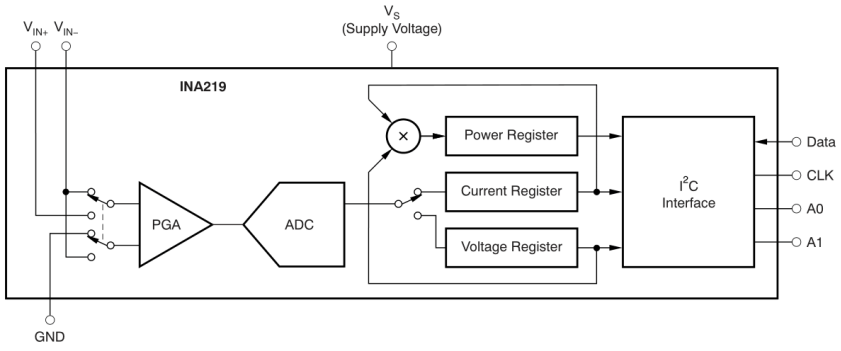
The INA219 is a high-side current shunt and power monitor with an I²C interface. The INA219 monitors both shunt drop and supply voltage, with programmable conversion times and filtering. A programmable calibration value, combined with an internal multiplier, enables direct readouts in amperes. An additional multiplying register calculates power in watts. The I²C interface features 16 programmable addresses.

The INA219 is available in two grades: A and B. The B grade version has higher accuracy and higher precision specifications.

The INA219 senses across shunts on buses that can vary from 0V to 26V. The device uses a single +3V to +5.5V supply, drawing a maximum of 1mA of supply current. The INA219 operates from –40°C to +125°C.

RELATED PRODUCTS

DESCRIPTION	DEVICE
Current/Power Monitor with Watchdog, Peak-Hold, and Fast Comparator Functions	INA209
Zerø-Drift, Low-Cost, Analog Current Shunt Monitor Series in Small Package	INA210, INA211, INA212, INA213, INA214



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

I²C is a trademark of NXP Semiconductors.

All other trademarks are the property of their respective owners.

This integrated circuit can be damaged by ESD. Texas Instruments recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage.

ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications.

Table 1. PACKAGING INFORMATION(1)

PRODUCT	PACKAGE-LEAD	PACKAGE DESIGNATOR	PACKAGE MARKING
INA219A	SO-8	D	I219A
	SOT23-8	DCN	A219
INA219B	SO-8	D	I219B
	SOT23-8	DCN	B219

(1) For the most current package and ordering information see the Package Option Addendum at the end of this document, or visit the INA219 product folder at www.ti.com.

ABSOLUTE MAXIMUM RATINGS(1)

Over operating free-air temperature range (unless otherwise noted).

		INA219	UNIT
Supply Voltage, V_S		6	V
Analog Inputs, V_{IN+}, V_{IN-}	Differential (V_{IN+}) – (V_{IN-})(2)	–26 to +26	V
	Common-Mode	-0.3 to +26	V
SDA		GND – 0.3 to +6	V
SCL		GND – 0.3 to V_S + 0.3	V
Input Current Into Any Pin		5	mA
Open-Drain Digital Output Current		10	mA
Operating Temperature		–40 to +125	°C
Storage Temperature		–65 to +150	°C
Junction Temperature		+150	°C
ESD Ratings	Human Body Model	4000	V
	Charged-Device Model	750	V
	Machine Model (MM)	200	V

(1) Stresses above these ratings may cause permanent damage. Exposure to absolute maximum conditions for extended periods may degrade device reliability. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those specified is not implied.
(2) V_{IN+} and V_{IN-} may have a differential voltage of –26V to +26V; however, the voltage at these pins must not exceed the range –0.3V to +26V.

ELECTRICAL CHARACTERISTICS: V_S = +3.3V

Boldface limits apply over the specified temperature range, **T_A = –25°C to +85°C.**
At T_A = +25°C, V_{IN+} = 12V, V_{SENSE} = (V_{IN+} – $V_{IN–}$) = 32mV, PGA = ÷ 1, and BRNG(1) = 1, unless otherwise noted.

PARAMETER		TEST CONDITIONS	INA219A MIN	INA219A TYP	INA219A MAX	INA219B MIN	INA219B TYP	INA219B MAX	UNIT
INPUT									
Full-Scale Current Sense (Input) Voltage Range		PGA = ÷ 1	0		±40	0		±40	mV
		PGA = ÷ 2	0		±80	0		±80	mV
		PGA = ÷ 4	0		±160	0		±160	mV
		PGA = ÷ 8	0		±320	0		±320	mV
Bus Voltage (Input Voltage) Range(2)		BRNG = 1	0		32	0		32	V
		BRNG = 0	0		16	0		16	V
Common-Mode Rejection	CMRR	V_{IN+} = 0V to 26V	100	120		100	120		dB
Offset Voltage, RTI(3)	V_{OS}	PGA = ÷ 1		±10	±100		±10	±50(4)	µV
		PGA = ÷ 2		±20	±125		±20	±75	µV
		PGA = ÷ 4		±30	±150		±30	±75	µV
		PGA = ÷ 8		±40	±200		±40	±100	µV
vs Temperature				**0.1**			**0.1**		**µV/°C**
vs Power Supply	PSRR	V_S = 3V to 5.5V		10			10		µV/V
Current Sense Gain Error				±40			±40		m%
vs Temperature				**1**			**1**		**m%/°C**
Input Impedance		Active Mode							
V_{IN+} Pin				20			20		µA
$V_{IN–}$ Pin				20 || 320			20 || 320		µA || kΩ
Input Leakage(5)		Power-Down Mode							
V_{IN+} Pin				0.1	±0.5		0.1	±0.5	µA
$V_{IN–}$ Pin				0.1	±0.5		0.1	±0.5	µA
DC ACCURACY									
ADC Basic Resolution				12			12		Bits
1 LSB Step Size									
Shunt Voltage				10			10		µV
Bus Voltage				4			4		mV
Current Measurement Error				±0.2	±0.5		±0.2	±0.3	%
over Temperature					**±1**			**±0.5**	**%**
Bus Voltage Measurement Error				±0.2	±0.5		±0.2	±0.5	%
over Temperature					**±1**			**±1**	**%**
Differential Nonlinearity				±0.1			±0.1		LSB
ADC TIMING									
ADC Conversion Time		12-Bit		532	586		532	586	µs
		11-Bit		276	304		276	304	µs
		10-Bit		148	163		148	163	µs
		9-Bit		84	93		84	93	µs
Minimum Convert Input Low Time			4			4			µs

(1) BRNG is bit 13 of the Configuration Register.
(2) This parameter only expresses the full-scale range of the ADC scaling. In no event should more than 26V be applied to this device.
(3) Referred-to-input (RTI).
(4) Shaded cells indicate improved specifications of the INA219B.
(5) Input leakage is positive (current flowing into the pin) for the conditions shown at the top of the table. Negative leakage currents can occur under different input conditions.

ELECTRICAL CHARACTERISTICS: V_S = +3.3V (continued)

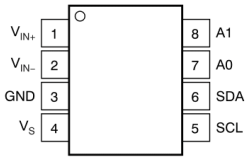
Boldface limits apply over the specified temperature range, **T_A = –25°C to +85°C.**
At T_A = +25°C, V_{IN+} = 12V, V_{SENSE} = (V_{IN+} – $V_{IN–}$) = 32mV, PGA = ÷ 1, and BRNG(1) = 1, unless otherwise noted.

PARAMETER	TEST CONDITIONS	INA219A			INA219B			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
SMBus								
SMBus Timeout(6)			28	35		28	35	ms
DIGITAL INPUTS (SDA as Input, SCL, A0, A1)								
Input Capacitance			3			3		pF
Leakage Input Current	$0 \le V_{IN} \le V_S$		0.1	1		0.1	1	µA
Input Logic Levels:								
V_{IH}		0.7 (V_S)		6	0.7 (V_S)		6	V
V_{IL}		–0.3		0.3 (V_S)	–0.3		0.3 (V_S)	V
Hysteresis			500			500		mV
OPEN-DRAIN DIGITAL OUTPUTS (SDA)								
Logic '0' Output Level	I_{SINK} = 3mA		0.15	0.4		0.15	0.4	V
High-Level Output Leakage Current	$V_{OUT} = V_S$		0.1	1		0.1	1	µA
POWER SUPPLY								
Operating Supply Range		+3		+5.5	+3		+5.5	V
Quiescent Current			0.7	1		0.7	1	mA
Quiescent Current, Power-Down Mode			6	15		6	15	µA
Power-On Reset Threshold			2			2		V
TEMPERATURE RANGE								
Specified Temperature Range		–25		+85	–25		+85	°C
Operating Temperature Range		–40		+125	–40		+125	°C
Thermal Resistance(7) θ_{JA}								
SOT23-8			142			142		°C/W
SO-8			120			120		°C/W

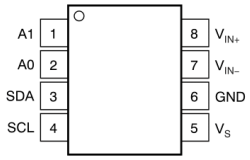
(6) SMBus timeout in the INA219 resets the interface any time SCL or SDA is low for over 28ms.
(7) θ_{JA} value is based on JEDEC low-K board.

PIN CONFIGURATIONS

DCN PACKAGE
SOT23-8
(Top View)



D PACKAGE
SO-8
(Top View)



PIN DESCRIPTIONS: SOT23-8

SOT23-8 (DCN)		DESCRIPTION
PIN NO	NAME	
1	V_{IN+}	Positive differential shunt voltage. Connect to positive side of shunt resistor.
2	V_{IN-}	Negative differential shunt voltage. Connect to negative side of shunt resistor. Bus voltage is measured from this pin to ground.
3	GND	Ground.
4	V_S	Power supply, 3V to 5.5V.
5	SCL	Serial bus clock line.
6	SDA	Serial bus data line.
7	A0	Address pin. Table 2 shows pin settings and corresponding addresses.
8	A1	Address pin. Table 2 shows pin settings and corresponding addresses.

PIN DESCRIPTIONS: SO-8

SO-8 (D)		DESCRIPTION
PIN NO	NAME	
1	A1	Address pin. Table 2 shows pin settings and corresponding addresses.
2	A0	Address pin. Table 2 shows pin settings and corresponding addresses.
3	SDA	Serial bus data line.
4	SCL	Serial bus clock line.
5	V_S	Power supply, 3V to 5.5V.
6	GND	Ground.
7	V_{IN-}	Negative differential shunt voltage. Connect to negative side of shunt resistor. Bus voltage is measured from this pin to ground.
8	V_{IN+}	Positive differential shunt voltage. Connect to positive side of shunt resistor.

TYPICAL CHARACTERISTICS

At T_A = +25°C, V_{IN+} = 12V, V_{SENSE} = (V_{IN+} – $V_{IN–}$) = 32mV, PGA = ÷ 1, and BRNG = 1, unless otherwise noted.

FREQUENCY RESPONSE

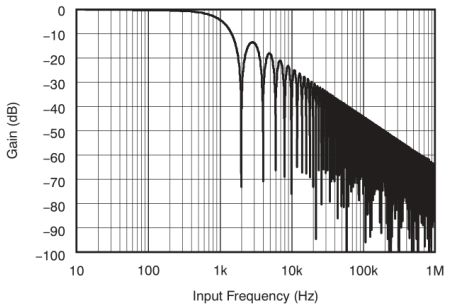


Figure 1.

ADC SHUNT OFFSET vs TEMPERATURE

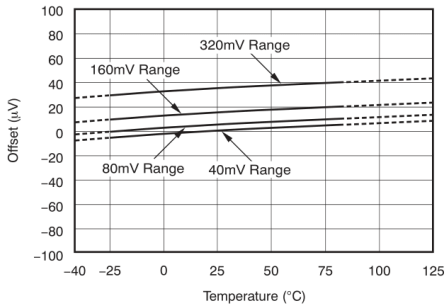


Figure 2.

ADC SHUNT GAIN ERROR vs TEMPERATURE

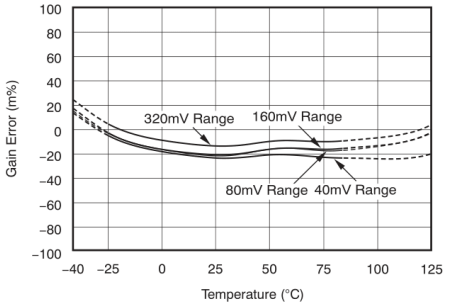


Figure 3.

ADC BUS VOLTAGE OFFSET vs TEMPERATURE

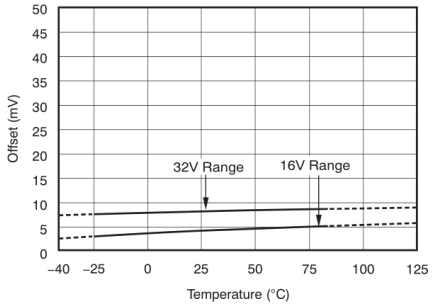


Figure 4.

ADC BUS GAIN ERROR vs TEMPERATURE

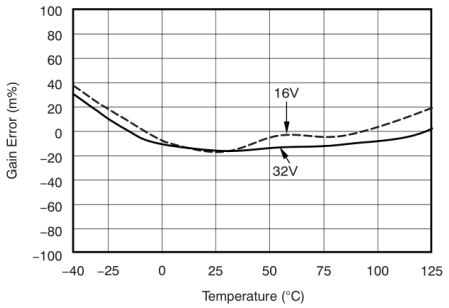


Figure 5.

INTEGRAL NONLINEARITY vs INPUT VOLTAGE

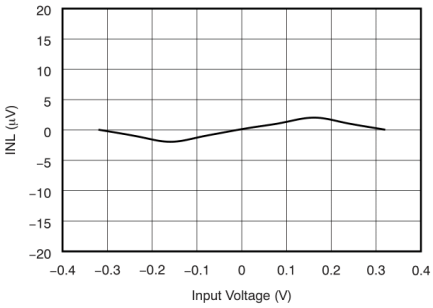


Figure 6.

TYPICAL CHARACTERISTICS (continued)

At T_A = +25°C, V_{IN+} = 12V, V_{SENSE} = (V_{IN+} – $V_{IN–}$) = 32mV, PGA = ÷ 1, and BRNG = 1, unless otherwise noted.

INPUT CURRENTS WITH LARGE DIFFERENTIAL VOLTAGES (V_{IN+} at 12V, Sweep of $V_{IN–}$)

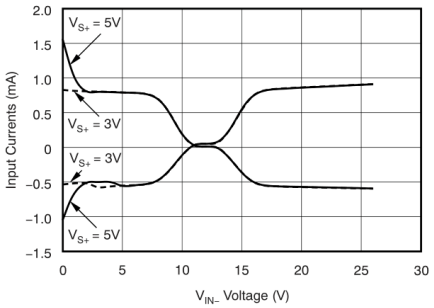


Figure 7.

ACTIVE I_Q vs TEMPERATURE

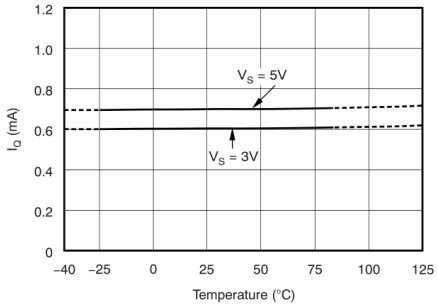


Figure 8.

SHUTDOWN I_Q vs TEMPERATURE

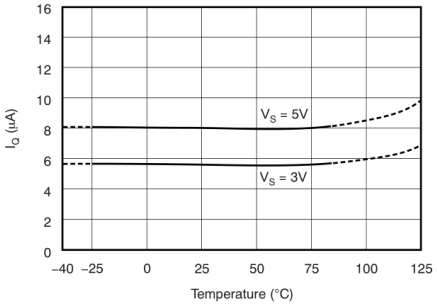


Figure 9.

ACTIVE I_Q vs I^2C CLOCK FREQUENCY

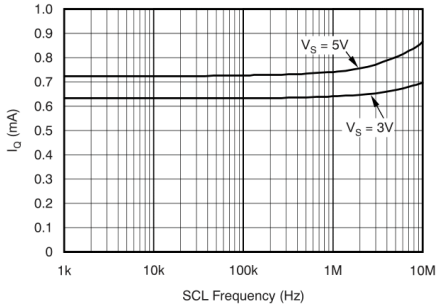


Figure 10.

SHUTDOWN I_Q vs I^2C CLOCK FREQUENCY

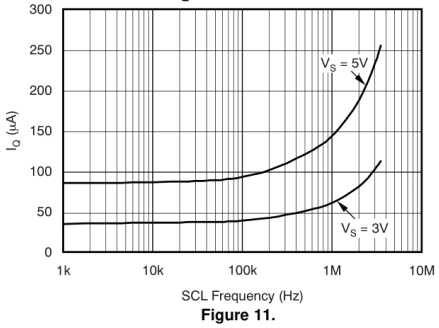


Figure 11.

REGISTER BLOCK DIAGRAM

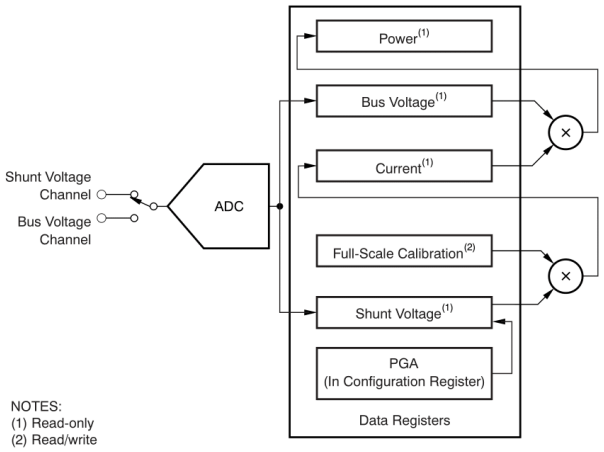


Figure 12. INA219 Register Block Diagram

APPLICATION INFORMATION

The INA219 is a digital current-shunt monitor with an I²C and SMBus-compatible interface. It provides digital current, voltage, and power readings necessary for accurate decision-making in precisely-controlled systems. Programmable registers allow flexible configuration for measurement resolution, and continuous-versus-triggered operation. Detailed register information appears at the end of this data sheet, beginning with Table 4. See the Register Block Diagram for a block diagram of the INA219.

INA219 TYPICAL APPLICATION

Figure 13 shows a typical application circuit for the INA219. Use a 0.1µF ceramic capacitor for power-supply bypassing, placed as closely as possible to the supply and ground pins.

The input filter circuit consisting of R_{F1}, R_{F2}, and C_F is not necessary in most applications. If the need for filtering is unknown, reserve board space for the components and install 0Ω resistors unless a filter is needed. See the *Filtering and Input Considerations* section.

The pull-up resistors shown on the SDA and SCL lines are not needed if there are pull-up resistors on these same lines elsewhere in the system. Resistor values shown are typical: consult either the I²C or SMBus specification to determine the acceptable minimum or maximum values.

BUS OVERVIEW

The INA219 offers compatibility with both I²C and SMBus interfaces. The I²C and SMBus protocols are essentially compatible with one another.

The I²C interface is used throughout this data sheet as the primary example, with SMBus protocol specified only when a difference between the two systems is being addressed. Two bidirectional lines, SCL and SDA, connect the INA219 to the bus. Both SCL and SDA are open-drain connections.

The device that initiates the transfer is called a *master*, and the devices controlled by the master are *slaves*. The bus must be controlled by a master device that generates the serial clock (SCL), controls the bus access, and generates START and STOP conditions.

To address a specific device, the master initiates a START condition by pulling the data signal line (SDA) from a HIGH to a LOW logic level while SCL is HIGH. All slaves on the bus shift in the slave address byte on the rising edge of SCL, with the last bit indicating whether a read or write operation is intended. During the ninth clock pulse, the slave being addressed responds to the master by generating an Acknowledge and pulling SDA LOW.

Data transfer is then initiated and eight bits of data are sent, followed by an *Acknowledge* bit. During data transfer, SDA must remain stable while SCL is HIGH. Any change in SDA while SCL is HIGH is interpreted as a START or STOP condition.

Once all data have been transferred, the master generates a STOP condition, indicated by pulling SDA from LOW to HIGH while SCL is HIGH. The INA219 includes a 28ms timeout on its interface to prevent locking up an SMBus.

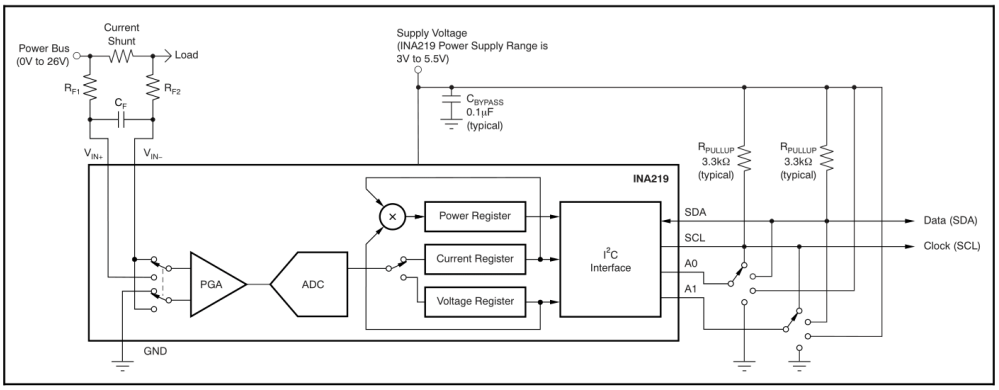


Figure 13. Typical Application Circuit

Serial Bus Address

To communicate with the INA219, the master must first address slave devices via a slave address byte. The slave address byte consists of seven address bits, and a direction bit indicating the intent of executing a read or write operation.

The INA219 has two address pins, A0 and A1. Table 2 describes the pin logic levels for each of the 16 possible addresses. The state of pins A0 and A1 is sampled on every bus communication and should be set before any activity on the interface occurs. The address pins are read at the start of each communication event.

Table 2. INA219 Address Pins and Slave Addresses

A1	A0	SLAVE ADDRESS
GND	GND	1000000
GND	V_{S+}	1000001
GND	SDA	1000010
GND	SCL	1000011
V_{S+}	GND	1000100
V_{S+}	V_{S+}	1000101
V_{S+}	SDA	1000110
V_{S+}	SCL	1000111
SDA	GND	1001000
SDA	V_{S+}	1001001
SDA	SDA	1001010
SDA	SCL	1001011
SCL	GND	1001100
SCL	V_{S+}	1001101
SCL	SDA	1001110
SCL	SCL	1001111

Serial Interface

The INA219 operates only as a slave device on the I^2C bus and SMBus. Connections to the bus are made via the open-drain I/O lines SDA and SCL. The SDA and SCL pins feature integrated spike suppression filters and Schmitt triggers to minimize the effects of input spikes and bus noise. The INA219 supports the transmission protocol for fast (1kHz to 400kHz) and high-speed (1kHz to 3.4MHz) modes. All data bytes are transmitted most significant byte first.

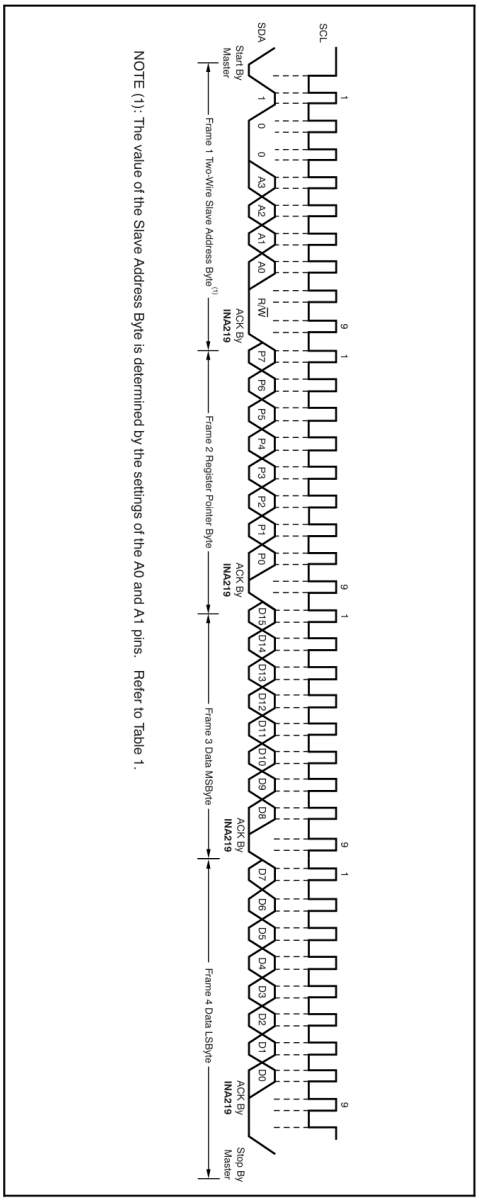
WRITING TO/READING FROM THE INA219

Accessing a particular register on the INA219 is accomplished by writing the appropriate value to the register pointer. Refer to Table 4 for a complete list of registers and corresponding addresses. The value for the register pointer as shown in Figure 17 is the first byte transferred after the slave address byte with the R/$\overline{W}$ bit LOW. Every write operation to the INA219 requires a value for the register pointer.

Writing to a register begins with the first byte transmitted by the master. This byte is the slave address, with the R/$\overline{W}$ bit LOW. The INA219 then acknowledges receipt of a valid address. The next byte transmitted by the master is the address of the register to which data will be written. This register address value updates the register pointer to the desired register. The next two bytes are written to the register addressed by the register pointer. The INA219 acknowledges receipt of each data byte. The master may terminate data transfer by generating a START or STOP condition.

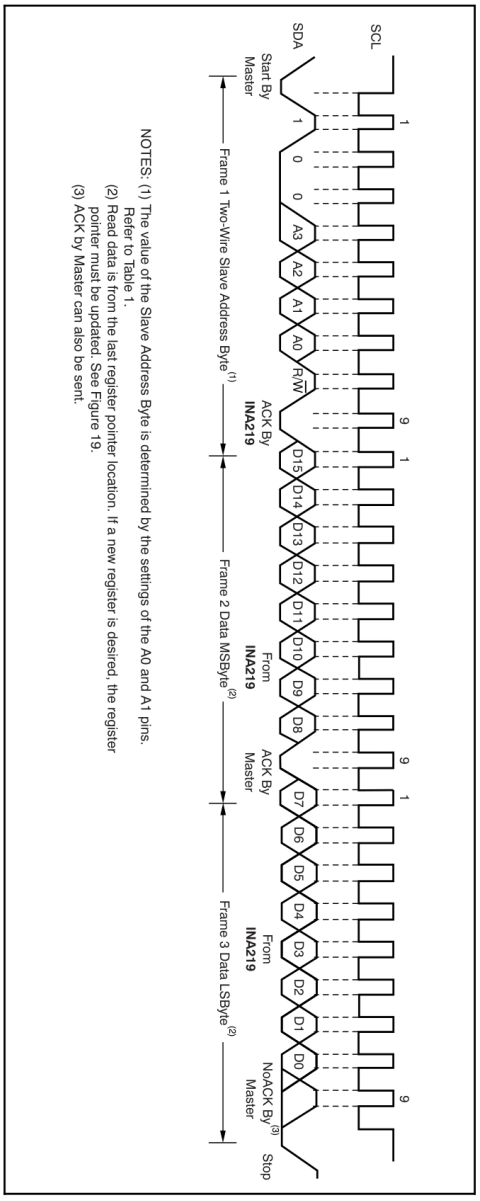
When reading from the INA219, the last value stored in the register pointer by a write operation determines which register is read during a read operation. To change the register pointer for a read operation, a new value must be written to the register pointer. This write is accomplished by issuing a slave address byte with the R/$\overline{W}$ bit LOW, followed by the register pointer byte. No additional data are required. The master then generates a START condition and sends the slave address byte with the R/$\overline{W}$ bit HIGH to initiate the read command. The next byte is transmitted by the slave and is the most significant byte of the register indicated by the register pointer. This byte is followed by an *Acknowledge* from the master; then the slave transmits the least significant byte. The master acknowledges receipt of the data byte. The master may terminate data transfer by generating a *Not-Acknowledge* after receiving any data byte, or generating a START or STOP condition. If repeated reads from the same register are desired, it is not necessary to continually send the register pointer bytes; the INA219 retains the register pointer value until it is changed by the next write operation.

Figure 14 and Figure 15 show read and write operation timing diagrams, respectively. Note that register bytes are sent most-significant byte first, followed by the least significant byte. Figure 16 shows the timing diagram for the SMBus Alert response operation. Figure 17 illustrates a typical register pointer configuration.



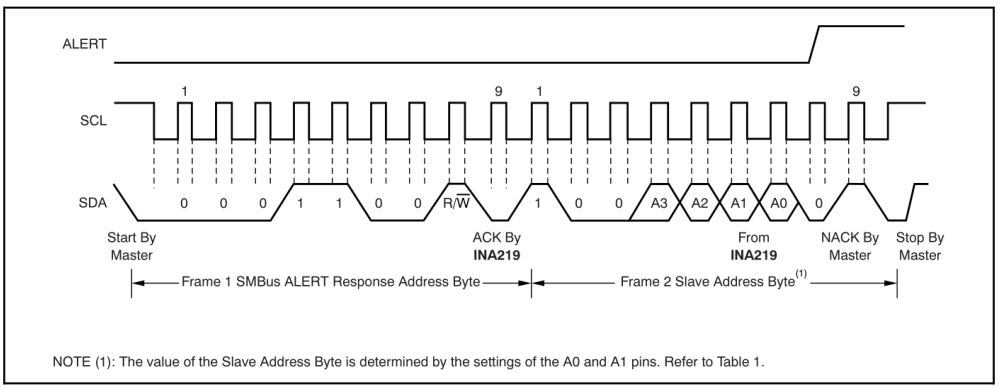
NOTE (1): The value of the Slave Address Byte is determined by the settings of the A0 and A1 pins. Refer to Table 1.

Figure 14. Timing Diagram for Write Word Format



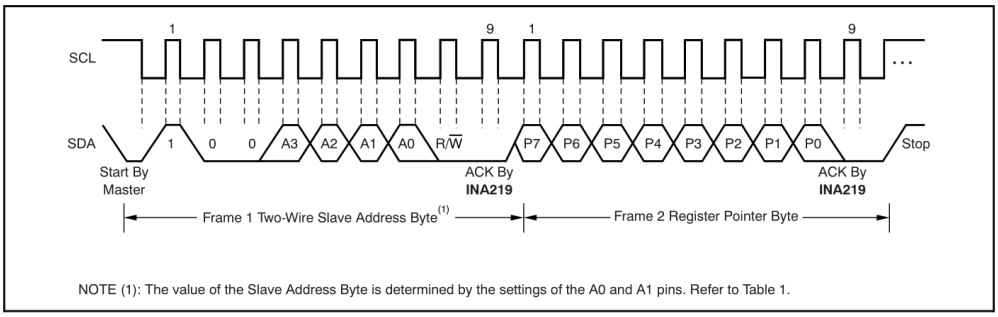
NOTES: (1) The value of the Slave Address Byte is determined by the settings of the A0 and A1 pins. Refer to Table 1.
(2) Read data is from the last register pointer location. If a new register is desired, the register pointer must be updated. See Figure 19.
(3) ACK by Master can also be sent.

Figure 15. Timing Diagram for Read Word Format



NOTE (1): The value of the Slave Address Byte is determined by the settings of the A0 and A1 pins. Refer to Table 1.

Figure 16. Timing Diagram for SMBus ALERT



NOTE (1): The value of the Slave Address Byte is determined by the settings of the A0 and A1 pins. Refer to Table 1.

Figure 17. Typical Register Pointer Set

High-Speed I²C Mode

When the bus is idle, both the SDA and SCL lines are pulled high by the pull-up devices. The master generates a start condition followed by a valid serial byte containing High-Speed (HS) master code *00001XXX*. This transmission is made in fast (400kbps) or standard (100kbps) (F/S) mode at no more than 400kbps. The INA219 does not acknowledge the HS master code, but does recognize it and switches its internal filters to support 3.4Mbps operation.

The master then generates a repeated start condition (a repeated start condition has the same timing as the start condition). After this repeated start condition, the protocol is the same as F/S mode, except that transmission speeds up to 3.4Mbps are allowed. Instead of using a stop condition, repeated start conditions should be used to secure the bus in HS-mode. A stop condition ends the HS-mode and switches all the internal filters of the INA219 to support the F/S mode.

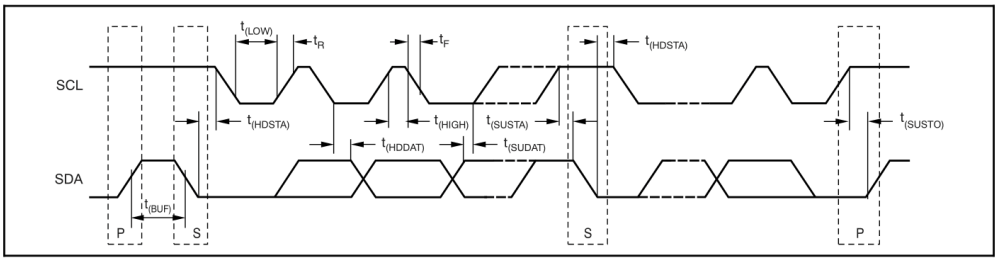


Figure 18. Bus Timing Diagram

Bus Timing Diagram Definitions

PARAMETER		FAST MODE		HIGH-SPEED MODE		UNITS
		MIN	MAX	MIN	MAX	
SCL Operating Frequency	$f_{(SCL)}$	0.001	0.4	0.001	3.4	MHz
Bus Free Time Between STOP and START Condition	$t_{(BUF)}$	600		160		ns
Hold time after repeated START condition. After this period, the first clock is generated.	$t_{(HDSTA)}$	100		100		ns
Repeated START Condition Setup Time	$t_{(SUSTA)}$	100		100		ns
STOP Condition Setup Time	$t_{(SUSTO)}$	100		100		ns
Data Hold Time	$t_{(HDDAT)}$	0		0		ns
Data Setup Time	$t_{(SUDAT)}$	100		10		ns
SCL Clock LOW Period	$t_{(LOW)}$	1300		160		ns
SCL Clock HIGH Period	$t_{(HIGH)}$	600		60		ns
Clock/Data Fall Time	t_F		300		160	ns
Clock/Data Rise Time	t_R		300		160	ns
Clock/Data Rise Time for SCLK ≤ 100kHz	t_R		1000			ns

Power-Up Conditions

Power-up conditions apply to a software reset via the RST bit (bit 15) in the Configuration Register, or the I^2C bus General Call Reset.

BASIC ADC FUNCTIONS

The two analog inputs to the INA219, V_{IN+} and V_{IN-}, connect to a shunt resistor in the bus of interest. The INA219 is typically powered by a separate supply from +3V to +5.5V. The bus being sensed can vary from 0V to 26V. There are no special considerations for power-supply sequencing (for example, a bus voltage can be present with the supply voltage off, and vice-versa). The INA219 senses the small drop across the shunt for shunt voltage, and senses the voltage with respect to ground from V_{IN-} for the bus voltage. Figure 19 illustrates this operation.

When the INA219 is in the normal operating mode (that is, MODE bits of the Configuration Register are set to '111'), it continuously converts the shunt voltage up to the number set in the shunt voltage averaging function (Configuration Register, SADC bits). The device then converts the bus voltage up to the number set in the bus voltage averaging (Configuration Register, BADC bits). The Mode control in the Configuration Register also permits selecting modes to convert only voltage or current, either continuously or in response to an event (triggered).

All current and power calculations are performed in the background and do not contribute to conversion time; conversion times shown in the Electrical Characteristics table can be used to determine the actual conversion time.

Power-Down mode reduces the quiescent current and turns off current into the INA219 inputs, avoiding any supply drain. Full recovery from Power-Down requires 40µs. ADC Off mode (set by the Configuration Register, MODE bits) stops all conversions.

Writing any of the triggered convert modes into the Configuration Register (even if the desired mode is already programmed into the register) triggers a single-shot conversion. Table 7 lists the triggered convert mode settings.

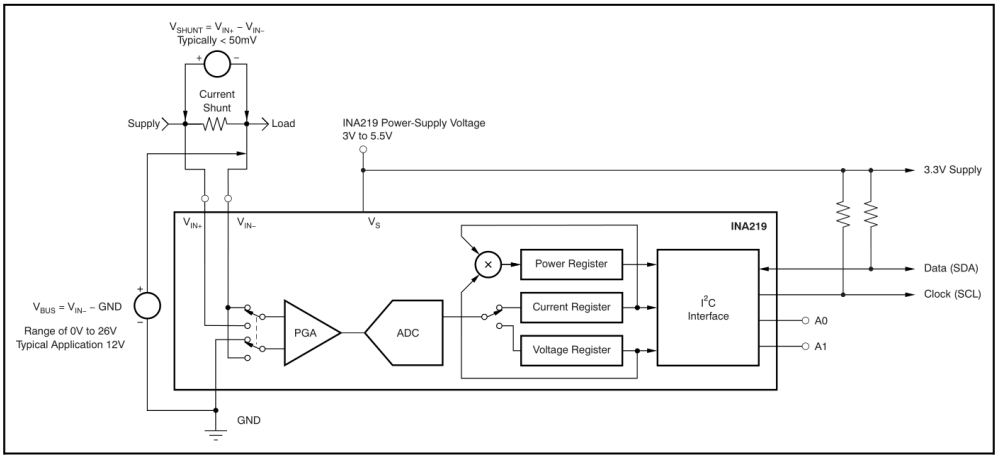


Figure 19. INA219 Configured for Shunt and Bus Voltage Measurement

Although the INA219 can be read at any time, and the data from the last conversion remain available, the Conversion Ready bit (Status Register, CNVR bit) is provided to help co-ordinate one-shot or triggered conversions. The Conversion Ready bit is set after all conversions, averaging, and multiplication operations are complete.

The Conversion Ready bit clears under these conditions:

1. Writing to the Configuration Register, except when configuring the MODE bits for Power Down or ADC off (Disable) modes;
2. Reading the Status Register; or
3. Triggering a single-shot conversion with the Convert pin.

Power Measurement

Current and bus voltage are converted at different points in time, depending on the resolution and averaging mode settings. For instance, when configured for 12-bit and 128 sample averaging, up to 68ms in time between sampling these two values is possible. Again, these calculations are performed in the background and do not add to the overall conversion time.

PGA Function

If larger full-scale shunt voltages are desired, the INA219 provides a PGA function that increases the full-scale range up to 2, 4, or 8 times (320mV). Additionally, the bus voltage measurement has two full-scale ranges: 16V or 32V.

Compatibility with TI Hot Swap Controllers

The INA219 is designed for compatibility with hot swap controllers such the TI TPS2490. The TPS2490 uses a high-side shunt with a limit at 50mV; the INA219 full-scale range of 40mV enables the use of the same shunt for current sensing below this limit. When sensing is required at (or through) the 50mV sense point of the TPS2490, the PGA of the INA219 can be set to ÷2 to provide an 80mV full-scale range.

Filtering and Input Considerations

Measuring current is often noisy, and such noise can be difficult to define. The INA219 offers several options for filtering by choosing resolution and averaging in the Configuration Register. These filtering options can be set independently for either voltage or current measurement.

The internal ADC is based on a delta-sigma (ΔΣ) front-end with a 500kHz (±30%) typical sampling rate. This architecture has good inherent noise rejection; however, transients that occur at or very close to the sampling rate harmonics can cause problems. Because these signals are at 1MHz and higher, they can be dealt with by incorporating filtering at the input of the INA219. The high frequency enables the use of low-value series resistors on the filter for negligible effects on measurement accuracy. In general, filtering the INA219 input is only necessary if there are transients at exact harmonics of the 500kHz (±30%) sampling rate (>1MHz). Filter using the lowest possible series resistance and ceramic capacitor. Recommended values are 0.1µF to 1.0µF. Figure 20 shows the INA219 with an additonal filter added at the input.

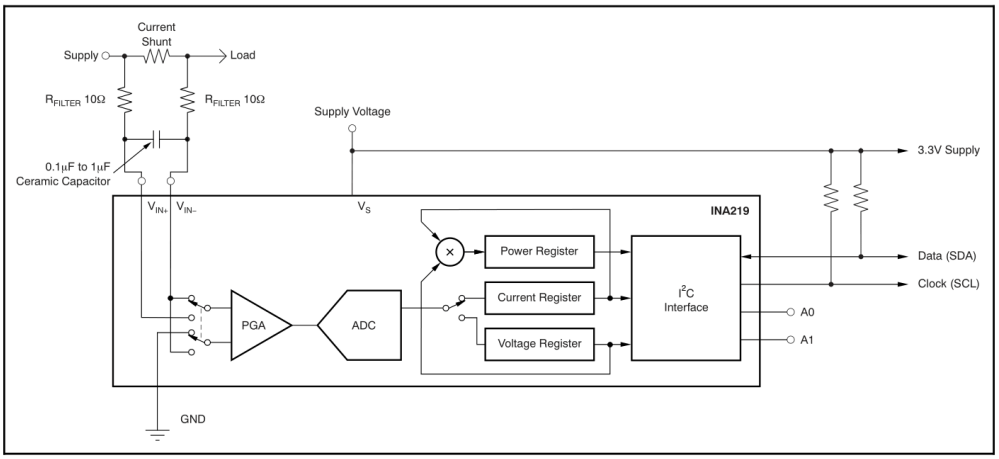


Figure 20. INA219 with Input Filtering

Overload conditions are another consideration for the INA219 inputs. The INA219 inputs are specified to tolerate 26V across the inputs. A large differential scenario might be a short to ground on the load side of the shunt. This type of event can result in full power-supply voltage across the shunt (as long the power supply or energy storage capacitors support it). It must be remembered that removing a short to ground can result in inductive kickbacks that could exceed the 26V differential and common-mode rating of the INA219. Inductive kickback voltages are best dealt with by zener-type transient-absorbing devices (commonly called *transzorbs*) combined with sufficient energy storage capacitance.

In applications that do not have large energy storage electrolytics on one or both sides of the shunt, an input overstress condition may result from an excessive dV/dt of the voltage applied to the input. A hard physical short is the most likely cause of this event, particularly in applications with no large electrolytics present. This problem occurs because an excessive dV/dt can activate the ESD protection in the INA219 in systems where large currents are available. Testing has demonstrated that the addition of 10Ω resistors in series with each input of the INA219 sufficiently protects the inputs against dV/dt failure up to the 26V rating of the INA219. These resistors have no significant effect on accuracy.

Simple Current Shunt Monitor Usage (No Programming Necessary)

The INA219 can be used without any programming if it is only necessary to read a shunt voltage drop and bus voltage with the default 12-bit resolution, 320mV shunt full-scale range (PGA = ÷8), 32V bus full-scale range, and continuous conversion of shunt and bus voltage.

Without programming, current is measured by reading the shunt voltage. The Current Register and Power Register are only available if the Calibration Register contains a programmed value.

Programming the INA219

The default power-up states of the registers are shown in the INA219 register descriptions section of this data sheet. These registers are volatile, and if programmed to other than default values, must be re-programmed at every device power-up. Detailed information on programming the Calibration Register specifically is given in the section, *Programming the INA219 Power Measurement Engine* .

PROGRAMMING THE INA219 POWER MEASUREMENT ENGINE

Calibration Register and Scaling

The Calibration Register makes it possible to set the scaling of the Current and Power Registers to whatever values are most useful for a given application. One strategy may be to set the Calibration Register such that the largest possible number is generated in the Current Register or Power Register at the expected full-scale point; this approach yields the highest resolution. The Calibration Register can also be selected to provide values in the Current and Power Registers that either provide direct decimal equivalents of the values being measured, or yield a round LSB number. After these choices have been made, the Calibration Register also offers possibilities for end user system-level calibration, where the value is adjusted slightly to cancel total system error.

Below are two examples for configuring the INA219 calibration. Both examples are written so the information directly relates to the calibration setup found in the INA219EVM software.

Calibration Example 1: Calibrating the INA219 with no possibility for overflow. (Note that the numbers used in this example are the same used with the INA219EVM software as shown in Figure 21.)

1. Establish the following parameters:

   V_{BUS_MAX} = 32

   V_{SHUNT_MAX} = 0.32

   R_{SHUNT} = 0.5

2. Using Equation 1, determine the maximum possible current .

$$MaxPossible\_I = \frac{V_{SHUNT\_MAX}}{R_{SHUNT}}$$

$$MaxPossible\_I = 0.64 \quad (1)$$

3. Choose the desired maximum current value. This value is selected based on system expectations.

   Max_Expected_I = 0.6

4. Calculate the possible range of current LSBs. To calculate this range, first compute a range of LSBs that is appropriate for the design. Next, select an LSB within this range. Note that the results will have the most resolution when the minimum LSB is selected. Typically, an LSB is selected to be the nearest round number to the minimum LSB value.

$$Minimum\_LSB = \frac{Max\_Expected\_I}{32767}$$

$$Minimum\_LSB = 18.311 \times 10^{-6} \quad (2)$$

$$Maximum\_LSB = \frac{Max\_Expected\_I}{4096}$$

$$Maximum\_LSB = 146.520 \times 10^{-6} \quad (3)$$

   Choose an LSB in the range: Minimum_LSB<Selected_LSB < Maximum_LSB

   $Current\_LSB = 20 \times 10^{-6}$

   Note:

   This value was selected to be a round number near the Minimum_LSB. This selection allows for good resolution with a rounded LSB.

5. Compute the Calibration Register value using Equation 4:

$$Cal = trunc\left(\frac{0.04096}{Current\_LSB \times R_{SHUNT}}\right)$$

$$Cal = 4096 \quad (4)$$

6. Calculate the Power LSB, using Equation 5. Equation 5 shows a general formula; because the bus voltage measurement LSB is always 4mV, the power formula reduces to the calculated result.

$$\text{Power\_LSB} = 20 \times \text{Current\_LSB}$$

$$\text{Power\_LSB} = 400 \times 10^{-6} \quad (5)$$

7. Compute the maximum current and shunt voltage values (before overflow), as shown by Equation 6 and Equation 7. Note that both Equation 6 and Equation 7 involve an *If - then* condition:

$$\text{Max\_Current} = \text{Current\_LSB} \times 32767$$

$$\text{Max\_Current} = 0.65534 \quad (6)$$

If Max_Current ≥ Max Possible_I then
Max_Current_Before_Overflow = MaxPossible_I
Else
Max_Current_Before_Overflow = Max_Current
End If

(Note that Max_Current is greater than MaxPossible_I in this example.)

Max_Current_Before_Overflow = 0.64 (Note: This result is displayed by software as seen in Figure 21.)

$$\text{Max\_ShuntVoltage} = \text{Max\_Current\_Before\_Overflow} \times R_{SHUNT}$$

$$\text{Max\_ShuntVoltage} = 0.32 \quad (7)$$

If Max_ShuntVoltage ≥ V_{SHUNT_MAX}
Max_ShuntVoltage_Before_Overflow = V_{SHUNT_MAX}
Else
Max_ShuntVoltage_Before_Overflow= Max_ShuntVoltage
End If

(Note that Max_ShuntVoltage is greater than V_{SHUNT_MAX} in this example.)

Max_ShuntVoltage_Before_Overflow = 0.32 (Note: This result is displayed by software as seen in Figure 21.)

8. Compute the maximum power with Equation 8.

$$\text{MaximumPower} = \text{Max\_Current\_Before\_Overflow} \times V_{BUS\_MAX}$$

$$\text{MaximumPower} = 20.48 \quad (8)$$

9. (Optional second Calibration step.) Compute corrected full-scale calibration value based on measured current.

INA219_Current = 0.63484

MeaShuntCurrent = 0.55

$$\text{Corrected\_Full\_Scale\_Cal} = \text{trunc}\left[\frac{\text{Cal} \times \text{MeasShuntCurrent}}{\text{INA219\_Current}}\right]$$

$$\text{Corrected\_Full\_Scale\_Cal} = 3548 \quad (9)$$

Figure 21 illustrates how to perform the same procedure discussed in this example using the automated INA219EVM software. Note that the same numbers used in the nine-step example are used in the software example in Figure 21. Also note that Figure 21 illustrates which results correspond to which step (for example, the information entered in Step 1 is enclosed in a box in Figure 21 and labeled).

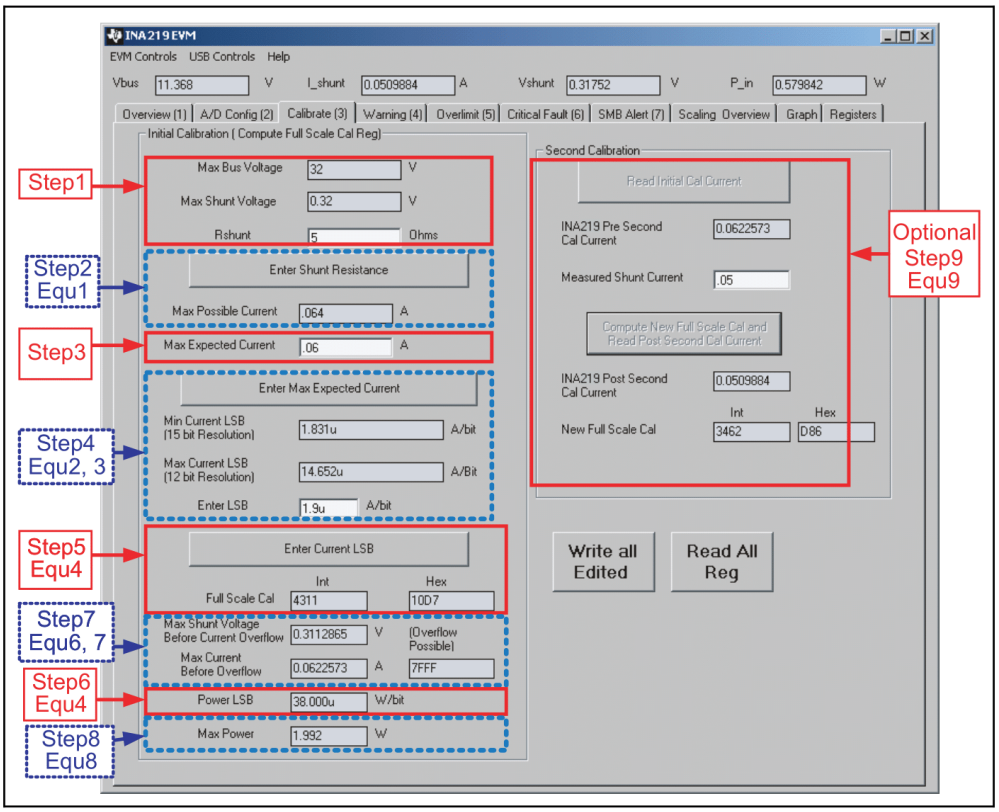


Figure 21. INA219 Calibration Sofware Automatically Computes Calibration Steps 1-9

Calibration Example 2 (Overflow Possible)

This design example uses the nine-step procedure for calibrating the INA219 where overflow is possible. Figure 22 illustrates how the same procedure is performed using the automated INA219EVM software. Note that the same numbers used in the nine-step example are used in the software example in Figure 22. Also note that Figure 22 illustrates which results correspond to which step (for example, the information entered in Step 1 is circled in Figure 22 and labeled).

1. Establish the following parameters:

   V_{BUS_MAX} = 32

   V_{SHUNT_MAX} = 0.32

   R_{SHUNT} = 5

2. Determine the maximum possible current using Equation 10:

$$MaxPossible\_I = \frac{V_{SHUNT\_MAX}}{R_{SHUNT}}$$

$$MaxPossible\_I = 0.064 \quad (10)$$

3. Choose the desired maximum current value: Max_Expected_I, ≤ MaxPossible_I. This value is selected based on system expectations.

   Max_Expected_I = 0.06

4. Calculate the possible range of current LSBs. This calculation is done by first computing a range of LSB's that is appropriate for the design. Next, select an LSB withing this range. Note that the results will have the most resolution when the minimum LSB is selected. Typically, an LSB is selected to be the nearest round number to the minimum LSB.

$$Minimum\_LSB = \frac{Max\_Expected\_I}{32767}$$

$$Minimum\_LSB = 1.831 \times 10^{-6} \quad (11)$$

$$Maximum\_LSB = \frac{Max\_Expected\_I}{4096}$$

$$Maximum\_LSB = 14.652 \times 10^{-6} \quad (12)$$

   Choose an LSB in the range: Minimum_LSB<Selected_LSB<Maximum_LSB

   $Current\_LSB = 1.9 \times 10^{-6}$

   Note:

   This value was selected to be a round number near the Minimum_LSB. This section allows for good resolution with a rounded LSB.

5. Compute the calibration register using Equation 13:

$$Cal = \text{trunc}\left(\frac{0.04096}{Current\_LSB \times R_{SHUNT}}\right) \quad Cal = 4311 \quad (13)$$

6. Calculate the Power LSB using Equation 14. Equation 14 shows a general formula; because the bus voltage measurement LSB is always 4mV, the power formula reduces to calculate the result.

$$Power\_LSB = 20\ Current\_LSB$$

$$Power\_LSB = 38 \times 10^{-6} \quad (14)$$

7. Compute the maximum current and shunt voltage values (before overflow), as shown by Equation 15 and Equation 16. Note that both Equation 15 and Equation 16 involve an *If - then* condition.

$$\text{Max\_Current} = \text{Current\_LSB} \times 32767$$
$$\text{Max\_Current} = 0.06226 \quad (15)$$

If Max_Current ≥ Max Possible_I then
Max_Current_Before_Overflow = MaxPossible_I
Else
Max_Current_Before_Overflow = Max_Current
End If

(Note that Max_Current is less than MaxPossible_I in this example.)

Max_Current_Before_Overflow = 0.06226 (Note: This result is displayed by software as seen in Figure 22.)

$$\text{Max\_ShuntVoltage} = \text{Max\_Current\_Before\_Overflow} \times R_{SHUNT}$$
$$\text{Max\_ShuntVoltage} = 0.3113 \quad (16)$$

If Max_ShuntVoltage ≥ V_{SHUNT_MAX}
Max_ShuntVoltage_Before_Overflow = V_{SHUNT_MAX}
Else
Max_ShuntVoltage_Before_Overflow= Max_ShuntVoltage
End If

(Note that Max_ShuntVoltage is less than V_{SHUNT_MAX} in this example.)

Max_ShuntVoltage_Before_Overflow = 0.3113 (Note: This result is displayed by software as seen in Figure 22.)

8. Compute the maximum power with equation 8.

$$\text{MaximumPower} = \text{Max\_Current\_Before\_Overflow} \times V_{BUS\_MAX}$$
$$\text{MaximumPower} = 1.992 \quad (17)$$

9. (Optional second calibration step.) Compute the corrected full-scale calibration value based on measured current.

INA219_Current = 0.06226

MeaShuntCurrent = 0.05

$$\text{Corrected\_Full\_Scale\_Cal} = \text{trunc}\left[\frac{\text{Cal} \times \text{MeasShuntCurrent}}{\text{INA219\_Current}}\right]$$
$$\text{Corrected\_Full\_Scale\_Cal} = 3462 \quad (18)$$

Figure 22 illustrates how to perform the same procedure discussed in this example using the automated INA219EVM software. Note that the same numbers used in the nine-step example are used in the software example in Figure 22.

Also note that Figure 22 illustrates which results correspond to which step (for example, the information entered in Step 1 is enclosed in a box in Figure 22 and labeled).

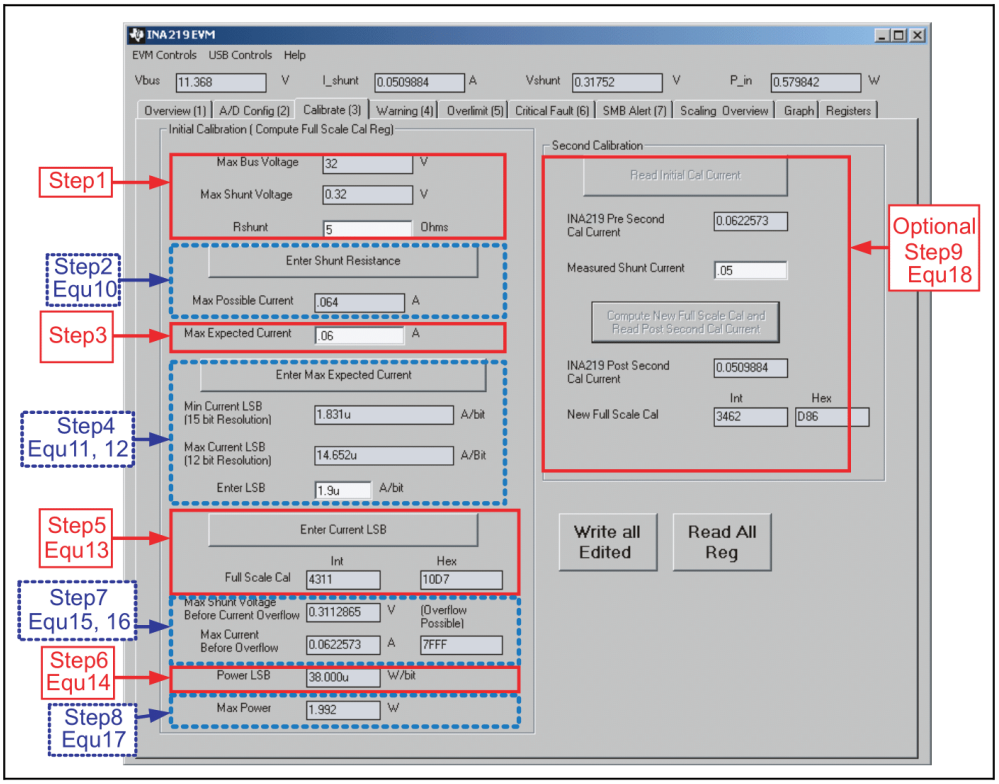


Figure 22. Calibration Software Automatically Computes Calibration Steps 1-9

CONFIGURE/MEASURE/CALCULATE EXAMPLE

In this example, the 10A load creates a differential voltage of 20mV across a 2mΩ shunt resistor. The voltage present at the V_{IN-} pin is equal to the common-mode voltage minus the differential drop across the resistor. The bus voltage for the INA219 is internally measured at the V_{IN-} pin to measure the voltage level delivered to the load. For this example, the voltage at the V_{IN-} pin is 11.98V. For this particular range (40mV full-scale), this small difference is not a significant deviation from the 12V common-mode voltage. However, at larger full-scale ranges, this deviation can be much larger.

Note that the Bus Voltage Register bits are not right-aligned. In order to compute the value of the Bus Voltage Register contents using the LSB of 4mV, the register must be shifted right by three bits. This shift puts the BD0 bit in the LSB position so that the contents can be multiplied by the 4mV LSB value to compute the bus voltage measured by the device. The shifted value of the bus voltage register contents is now equal to BB3h, a decimal equivalent of 2995. This value of 2995 multiplied by the 4mV LSB results in a value of 11.98V.

The Calibration Register (05h) is set in order to provide the device information about the current shunt resistor that was used to create the measured shunt voltage. By knowing the value of the shunt resistor, the device can then calculate the amount of current that created the measured shunt voltage drop. The first step when calculating the calibration value is setting the current LSB. The Calibration Register value is based on a calculation that has its precision capability limited by the size of the register and the Current Register LSB. The device can measure bidirectional current; thus, the MSB of the Current Register is a sign bit that allows for the rest of the 15 bits to be used for the Current Register value. It is common when using the current value calculations to use a resolution between 12 bits and 15 bits. Calculating the current LSB for each of these resolutions provides minimum and maximum values. These values are calculated assuming the maximum current that will be expected to flow through the current shunt resistor, as shown in Equation 2 and Equation 3. To simplify the mathematics, it is common to choose a round number located between these two points. For this example, the maximum current LSB is 3.66mA/bit and the minimum current LSB would be 457.78μA/bit assuming a maximum expected current of 15A. For this example, a value of 1mA/bit was chosen for the current LSB. Setting the current LSB to this value allows for sufficient precision while serving to simplify the math as well. Using Equation 4 results in a Calibration Register value of 20480, or 5000h.

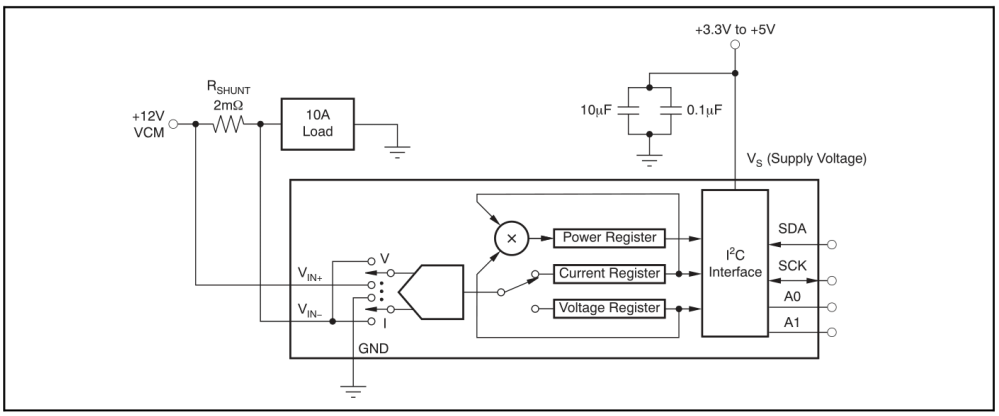


Figure 23. Example Circuit Configuration

The Current Register (04h) is then calculated by multiplying the shunt voltage contents by the Calibration Register and then dividing by 4096. For this example, the shunt voltage of 2000 is multiplied by the calibration register of 20480 and then divided by 4096 to yield a Current Register of 2710h.

The Power Register (03h) is then be calculated by multiplying the Current Register of 10000 by the Bus Voltage Register of 2995 and then dividing by 5000. For this example, the Power Register contents are 1766h, or a decimal equivalent of 5990. Multiplying this result by the power LSB that is 20 times the 1×10^{-3} current LSB, or 20×10^{-3}, results in a power calculation of 5990 × 20mW/bit, which equals 119.8W. This result matches what is expected for this register. A manual calculation for the power being delivered to the load would use 11.98V (12VCM – 20mV shunt drop) multiplied by the load current of 10A to give a 119.8W result.

Table 3 shows the steps for configuring, measuring, and calculating the values for current and power for this device.

Table 3. Configure/Measure/Calculate Example(1)

STEP #	REGISTER NAME	ADDRESS	CONTENTS	ADJ	DEC	LSB	VALUE
Step 1	Configuration	00h	019Fh				
Step 2	Shunt	01h	07D0h		2000	10µV	20mV
Step 3	Bus	02h	5D98h	0BB3	2995	4mV	11.98V
Step 4	Calibration	05h	5000h		20480		
Step 5	Current	04h	2710h		10000	1mA	10.0A
Step 6	Power	03h	1766h		5990	20mW	119.8W

(1) Conditions: load = 10A, V_{CM} = 12V, R_{SHUNT} = 2mΩ, V_{SHUNT} FSR = 40mV, and V_{BUS} = 16V.

REGISTER INFORMATION

The INA219 uses a bank of registers for holding configuration settings, measurement results, maximum/minimum limits, and status information. Table 4 summarizes the INA219 registers; Figure 12 illustrates registers.

Register contents are updated 4μs after completion of the write command. Therefore, a 4μs delay is required between completion of a write to a given register and a subsequent read of that register (without changing the pointer) when using SCL frequencies in excess of 1MHz.

Table 4. Summary of Register Set

POINTER ADDRESS			POWER-ON RESET		
HEX	REGISTER NAME	FUNCTION	BINARY	HEX	TYPE(1)
00	Configuration Register	All-register reset, settings for bus voltage range, PGA Gain, ADC resolution/averaging.	00111001 10011111	399F	R/$\overline{W}$
01	Shunt Voltage	Shunt voltage measurement data.	Shunt voltage	—	R
02	Bus Voltage	Bus voltage measurement data.	Bus voltage	—	R
03	Power(2)	Power measurement data.	00000000 00000000	0000	R
04	Current(2)	Contains the value of the current flowing through the shunt resistor.	00000000 00000000	0000	R
05	Calibration	Sets full-scale range and LSB of current and power measurements. Overall system calibration.	00000000 00000000	0000	R/$\overline{W}$

(1) Type: **R** = Read-Only, **R/$\overline{W}$** = Read/Write.
(2) The Power Register and Current Register default to '0' because the Calibration Register defaults to '0', yielding a zero current value until the Calibration Register is programmed.

REGISTER DETAILS

All INA219 registers 16-bit registers are actually two 8-bit bytes via the I^2C interface.

Configuration Register 00h (Read/Write)

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	RST	—	BRNG	PG1	PG0	BADC4	BADC3	BADC2	BADC1	SADC4	SADC3	SADC2	SADC1	MODE3	MODE2	MODE1
POR VALUE	0	0	1	1	1	0	0	1	1	0	0	1	1	1	1	1

Bit Descriptions

RST: **Reset Bit**

Bit 15 Setting this bit to '1' generates a system reset that is the same as power-on reset. Resets all registers to default values; this bit self-clears.

BRNG: **Bus Voltage Range**

Bit 13
0 = 16V FSR
1 = 32V FSR (default value)

PG: **PGA (Shunt Voltage Only)**

Bits 11, 12 Sets PGA gain and range. Note that the PGA defaults to ÷8 (320mV range). Table 5 shows the gain and range for the various product gain settings.

Table 5. PG Bit Settings(1)

PG1	PG0	GAIN	RANGE
0	0	1	±40mV
0	1	÷2	±80mV
1	0	÷4	±160mV
1	1	÷8	±320mV

(1) Shaded values are default.

BADC: **BADC Bus ADC Resolution/Averaging**

Bits 7–10 These bits adjust the Bus ADC resolution (9-, 10-, 11-, or 12-bit) or set the number of samples used when averaging results for the Bus Voltage Register (02h).

SADC: **SADC Shunt ADC Resolution/Averaging**

Bits 3–6 These bits adjust the Shunt ADC resolution (9-, 10-, 11-, or 12-bit) or set the number of samples used when averaging results for the Shunt Voltage Register (01h).
BADC (Bus) and SADC (Shunt) ADC resolution/averaging and conversion time settings are shown in Table 6.

Table 6. ADC Settings(1)

ADC4	ADC3	ADC2	ADC1	MODE/SAMPLES	CONVERSION TIME
0	X(2)	0	0	9-bit	84µs
0	X(2)	0	1	10-bit	148µs
0	X(2)	1	0	11-bit	276µs
0	X(2)	1	1	12-bit	532µs
1	0	0	0	12-bit	532µs
1	0	0	1	2	1.06ms
1	0	1	0	4	2.13ms
1	0	1	1	8	4.26ms
1	1	0	0	16	8.51ms
1	1	0	1	32	17.02ms
1	1	1	0	64	34.05ms
1	1	1	1	128	68.10ms

(1) Shaded values are default.
(2) X = Don't care.

MODE: **Operating Mode**

Bits 0–2 Selects continuous, triggered, or power-down mode of operation. These bits default to continuous shunt and bus measurement mode. The mode settings are shown in Table 7.

Table 7. Mode Settings(1)

MODE3	MODE2	MODE1	MODE
0	0	0	Power-Down
0	0	1	Shunt Voltage, Triggered
0	1	0	Bus Voltage, Triggered
0	1	1	Shunt and Bus, Triggered
1	0	0	ADC Off (disabled)
1	0	1	Shunt Voltage, Continuous
1	1	0	Bus Voltage, Continuous
1	1	1	Shunt and Bus, Continuous

(1) Shaded values are default.

DATA OUTPUT REGISTERS

Shunt Voltage Register 01h (Read-Only)

The Shunt Voltage Register stores the current shunt voltage reading, V_{SHUNT}. Shunt Voltage Register bits are shifted according to the PGA setting selected in the Configuration Register (00h). When multiple sign bits are present, they will all be the same value. Negative numbers are represented in two's complement format. Generate the two's complement of a negative number by complementing the absolute value binary number and adding 1. Extend the sign, denoting a negative number by setting the MSB = '1'. Extend the sign to any additional sign bits to form the 16-bit word.

Example: For a value of V_{SHUNT} = –320mV:

1. Take the absolute value (include accuracy to 0.01mV)==> 320.00
2. Translate this number to a whole decimal number ==> 32000
3. Convert it to binary==> 111 1101 0000 0000
4. Complement the binary result : 000 0010 1111 1111
5. Add 1 to the Complement to create the Two's Complement formatted result ==> 000 0011 0000 0000
6. Extend the sign and create the 16-bit word: 1000 0011 0000 0000 = 8300h (Remember to extend the sign to all sign-bits, as necessary based on the PGA setting.)

At PGA = ÷8, full-scale range = ±320mV (decimal = 32000, positive value hex = 7D00, negative value hex = 8300), and LSB = 10µV.

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	SIGN	SD14_8	SD13_8	SD12_8	SD11_8	SD10_8	SD9_8	SD8_8	SD7_8	SD6_8	SD5_8	SD4_8	SD3_8	SD2_8	SD1_8	SD0_8
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

At PGA = ÷4, full-scale range = ±160mV (decimal = 16000, positive value hex = 3E80, negative value hex = C180), and LSB = 10µV.

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	SIGN	SIGN	SD13_4	SD12_4	SD11_4	SD10_4	SD9_4	SD8_4	SD7_4	SD6_4	SD5_4	SD4_4	SD3_4	SD2_4	SD1_4	SD0_4
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

At PGA = ÷2, full-scale range = ±80mV (decimal = 8000, positive value hex = 1F40, negative value hex = E0C0), and LSB = 10µV.

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	SIGN	SIGN	SIGN	SD12_2	SD11_2	SD10_2	SD9_2	SD8_2	SD7_2	SD6_2	SD5_2	SD4_2	SD3_2	SD2_2	SD1_2	SD0_2
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

At PGA = ÷1, full-scale range = ±40mV (decimal = 4000, positive value hex = 0FA0, negative value hex = F060), and LSB = 10µV.

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	SIGN	SIGN	SIGN	SIGN	SD11_1	SD10_1	SD9_1	SD8_1	SD7_1	SD6_1	SD5_1	SD4_1	SD3_1	SD2_1	SD1_1	SD0_1
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8. Shunt Voltage Register Format(1)

V_{SHUNT} Reading (mV)	Decimal Value	PGA = ÷ 8 (D15.....D0)	PGA = ÷ 4 (D15.....D0)	PGA = ÷ 2 (D15.....D0)	PGA = ÷ 1 (D15.....D0)
320.02	32002	0111 1101 0000 0000	0011 1110 1000 0000	0001 1111 0100 0000	0000 1111 1010 0000
320.01	32001	0111 1101 0000 0000	0011 1110 1000 0000	0001 1111 0100 0000	0000 1111 1010 0000
320.00	32000	0111 1101 0000 0000	0011 1110 1000 0000	0001 1111 0100 0000	0000 1111 1010 0000
319.99	31999	0111 1100 1111 1111	0011 1110 1000 0000	0001 1111 0100 0000	0000 1111 1010 0000
319.98	31998	0111 1100 1111 1110	0011 1110 1000 0000	0001 1111 0100 0000	0000 1111 1010 0000
⋮	⋮	⋮	⋮	⋮	⋮
160.02	16002	0011 1110 1000 0010	0011 1110 1000 0000	0001 1111 0100 0000	0000 1111 1010 0000
160.01	16001	0011 1110 1000 0001	0011 1110 1000 0000	0001 1111 0100 0000	0000 1111 1010 0000
160.00	16000	0011 1110 1000 0000	0011 1110 1000 0000	0001 1111 0100 0000	0000 1111 1010 0000
159.99	15999	0011 1110 0111 1111	0011 1110 0111 1111	0001 1111 0100 0000	0000 1111 1010 0000
159.98	15998	0011 1110 0111 1110	0011 1110 0111 1110	0001 1111 0100 0000	0000 1111 1010 0000
⋮	⋮	⋮	⋮	⋮	⋮
80.02	8002	0001 1111 0100 0010	0001 1111 0100 0010	0001 1111 0100 0000	0000 1111 1010 0000
80.01	8001	0001 1111 0100 0001	0001 1111 0100 0001	0001 1111 0100 0000	0000 1111 1010 0000
80.00	8000	0001 1111 0100 0000	0001 1111 0100 0000	0001 1111 0100 0000	0000 1111 1010 0000
79.99	7999	0001 1111 0011 1111	0001 1111 0011 1111	0001 1111 0011 1111	0000 1111 1010 0000
79.98	7998	0001 1111 0011 1110	0001 1111 0011 1110	0001 1111 0011 1110	0000 1111 1010 0000
⋮	⋮	⋮	⋮	⋮	⋮
40.02	4002	0000 1111 1010 0010	0000 1111 1010 0010	0000 1111 1010 0010	0000 1111 1010 0000
40.01	4001	0000 1111 1010 0001	0000 1111 1010 0001	0000 1111 1010 0001	0000 1111 1010 0000
40.00	4000	0000 1111 1010 0000	0000 1111 1010 0000	0000 1111 1010 0000	0000 1111 1010 0000
39.99	3999	0000 1111 1001 1111	0000 1111 1001 1111	0000 1111 1001 1111	0000 1111 1001 1111
39.98	3998	0000 1111 1001 1110	0000 1111 1001 1110	0000 1111 1001 1110	0000 1111 1001 1110
⋮	⋮	⋮	⋮	⋮	⋮
0.02	2	0000 0000 0000 0010	0000 0000 0000 0010	0000 0000 0000 0010	0000 0000 0000 0010
0.01	1	0000 0000 0000 0001	0000 0000 0000 0001	0000 0000 0000 0001	0000 0000 0000 0001
0	0	0000 0000 0000 0000	0000 0000 0000 0000	0000 0000 0000 0000	0000 0000 0000 0000
–0.01	–1	1111 1111 1111 1111	1111 1111 1111 1111	1111 1111 1111 1111	1111 1111 1111 1111
–0.02	–2	1111 1111 1111 1110	1111 1111 1111 1110	1111 1111 1111 1110	1111 1111 1111 1110
⋮	⋮	⋮	⋮	⋮	⋮
–39.98	–3998	1111 0000 0110 0010	1111 0000 0110 0010	1111 0000 0110 0010	1111 0000 0110 0010
–39.99	–3999	1111 0000 0110 0001	1111 0000 0110 0001	1111 0000 0110 0001	1111 0000 0110 0001
–40.00	–4000	1111 0000 0110 0000	1111 0000 0110 0000	1111 0000 0110 0000	1111 0000 0110 0000
–40.01	–4001	1111 0000 0101 1111	1111 0000 0101 1111	1111 0000 0101 1111	1111 0000 0110 0000
–40.02	–4002	1111 0000 0101 1110	1111 0000 0101 1110	1111 0000 0101 1110	1111 0000 0110 0000
⋮	⋮	⋮	⋮	⋮	⋮
–79.98	–7998	1110 0000 1100 0010	1110 0000 1100 0010	1110 0000 1100 0010	1111 0000 0110 0000
–79.99	–7999	1110 0000 1100 0001	1110 0000 1100 0001	1110 0000 1100 0001	1111 0000 0110 0000
–80.00	–8000	1110 0000 1100 0000	1110 0000 1100 0000	1110 0000 1100 0000	1111 0000 0110 0000
–80.01	–8001	1110 0000 1011 1111	1110 0000 1011 1111	1110 0000 1100 0000	1111 0000 0110 0000
–80.02	–8002	1110 0000 1011 1110	1110 0000 1011 1110	1110 0000 1100 0000	1111 0000 0110 0000
⋮	⋮	⋮	⋮	⋮	⋮
–159.98	–15998	1100 0001 1000 0010	1100 0001 1000 0010	1110 0000 1100 0000	1111 0000 0110 0000
–159.99	–15999	1100 0001 1000 0001	1100 0001 1000 0001	1110 0000 1100 0000	1111 0000 0110 0000
–160.00	–16000	1100 0001 1000 0000	1100 0001 1000 0000	1110 0000 1100 0000	1111 0000 0110 0000
–160.01	–16001	1100 0001 0111 1111	1100 0001 1000 0000	1110 0000 1100 0000	1111 0000 0110 0000
–160.02	–16002	1100 0001 0111 1110	1100 0001 1000 0000	1110 0000 1100 0000	1111 0000 0110 0000
⋮	⋮	⋮	⋮	⋮	⋮
–319.98	–31998	1000 0011 0000 0010	1100 0001 1000 0000	1110 0000 1100 0000	1111 0000 0110 0000
–319.99	–31999	1000 0011 0000 0001	1100 0001 1000 0000	1110 0000 1100 0000	1111 0000 0110 0000
–320.00	–32000	1000 0011 0000 0000	1100 0001 1000 0000	1110 0000 1100 0000	1111 0000 0110 0000
–320.01	–32001	1000 0011 0000 0000	1100 0001 1000 0000	1110 0000 1100 0000	1111 0000 0110 0000
–320.02	–32002	1000 0011 0000 0000	1100 0001 1000 0000	1110 0000 1100 0000	1111 0000 0110 0000

(1) Out-of-range values are shown in grey shading.

Bus Voltage Register 02h (Read-Only)

The Bus Voltage Register stores the most recent bus voltage reading, V_{BUS}.

At full-scale range = 32V (decimal = 8000, hex = 1F40), and LSB = 4mV.

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	BD12	BD11	BD10	BD9	BD8	BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0	—	CNVR	OVF
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

At full-scale range = 16V (decimal = 4000, hex = 0FA0), and LSB = 4mV.

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	0	BD11	BD10	BD9	BD8	BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0	—	CNVR	OVF
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CNVR: **Conversion Ready**

Bit 1

Although the data from the last conversion can be read at any time, the INA219 Conversion Ready bit (CNVR) indicates when data from a conversion is available in the data output registers. The CNVR bit is set after all conversions, averaging, and multiplications are complete. CNVR will clear under the following conditions:

1.) Writing a new mode into the Operating Mode bits in the Configuration Register (except for Power-Down or Disable)

2.) Reading the Power Register

OVF: **Math Overflow Flag**

Bit 0

The Math Overflow Flag (OVF) is set when the Power or Current calculations are out of range. It indicates that current and power data may be meaningless.

Power Register 03h (Read-Only)

Full-scale range and LSB are set by the Calibration Register. See the *Programming the INA219 Power Measurement Engine* section.

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The Power Register records power in watts by multiplying the values of the current with the value of the bus voltage according to the equation:

$$\text{Power} = \frac{\text{Current} \times \text{BusVoltage}}{5000}$$

Current Register 04h (Read-Only)

Full-scale range and LSB depend on the value entered in the Calibration Register. See the *Programming the INA219 Power Measurement Engine* section. Negative values are stored in two's complement format.

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	CSIGN	CD14	CD13	CD12	CD11	CD10	CD9	CD8	CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The value of the Current Register is calculated by multiplying the value in the Shunt Voltage Register with the value in the Calibration Register according to the equation:

$$\text{Current} = \frac{\text{ShuntVoltage} \times \text{Calibration Register}}{4096}$$

CALIBRATION REGISTER

Calibration Register 05h (Read/Write)

Current and power calibration are set by bits D15 to D1 of the Calibration Register. Note that bit D0 is not used in the calculation. This register sets the current that corresponds to a full-scale drop across the shunt. Full-scale range and the LSB of the current and power measurement depend on the value entered in this register. See the *Programming the INA219 Power Measurement Engine* section. This register is suitable for use in overall system calibration. Note that the '0' POR values are all default.

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0(1)
BIT NAME	FS15	FS14	FS13	FS12	FS11	FS10	FS9	FS8	FS7	FS6	FS5	FS4	FS3	FS2	FS1	FS0
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(1) D0 is a *void* bit and will always be '0'. It is not possible to write a '1' to D0. CALIBRATION is the value stored in D15:D1.

REVISION HISTORY

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision E (September 2010) to Revision F **Page**

- Changed values in text. 24
- Changed step 5 and step 6 values in Table 3 24

Changes from Revision D (September 2010) to Revision E **Page**

- Updated *Packaging Information* table 2

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/ Ball Finish	MSL Peak Temp (3)	Samples (Requires Login)
INA219AID	ACTIVE	SOIC	D	8	75	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	
INA219AIDCNR	ACTIVE	SOT-23	DCN	8	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	
INA219AIDCNRG4	ACTIVE	SOT-23	DCN	8	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	
INA219AIDCNT	ACTIVE	SOT-23	DCN	8	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	
INA219AIDCNTG4	ACTIVE	SOT-23	DCN	8	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	
INA219AIDR	ACTIVE	SOIC	D	8	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	
INA219BID	ACTIVE	SOIC	D	8	75	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	
INA219BIDCNR	ACTIVE	SOT-23	DCN	8	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	
INA219BIDCNT	ACTIVE	SOT-23	DCN	8	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	
INA219BIDR	ACTIVE	SOIC	D	8	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM	

(1) The marketing status values are defined as follows:
ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

(2) Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check http://www.ti.com/productcontent for the latest availability information and additional product content details.
TBD: The Pb-Free/Green conversion plan has not been defined.
Pb-Free (RoHS): TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.
Pb-Free (RoHS Exempt): This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.

PACKAGE OPTION ADDENDUM

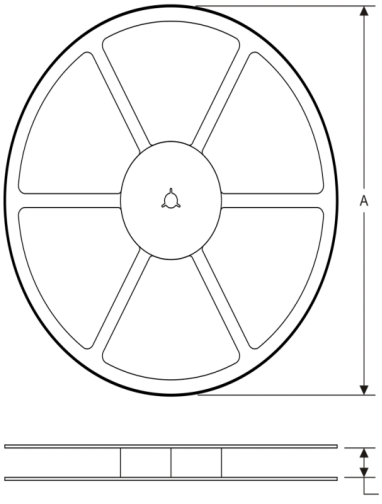
www.ti.com 22-Sep-2011

Green (RoHS & no Sb/Br): TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

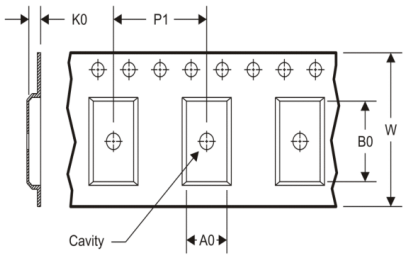
(3) MSL, Peak Temp. -- The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

TAPE AND REEL INFORMATION

REEL DIMENSIONS



TAPE DIMENSIONS



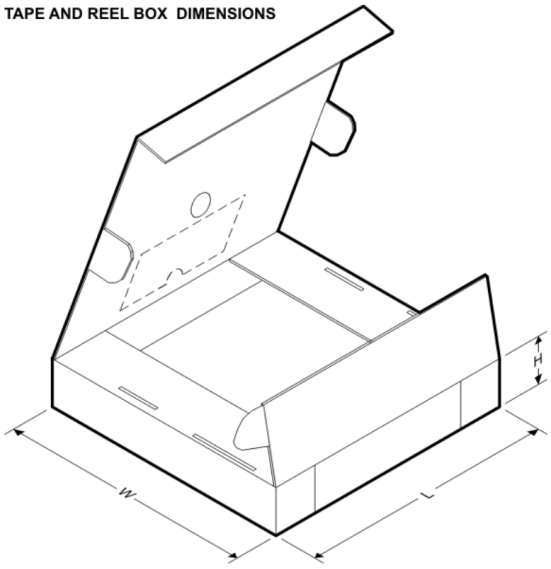
A0	Dimension designed to accommodate the component width
B0	Dimension designed to accommodate the component length
K0	Dimension designed to accommodate the component thickness
W	Overall width of the carrier tape
P1	Pitch between successive cavity centers

TAPE AND REEL INFORMATION

*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
INA219AIDCNR	SOT-23	DCN	8	3000	179.0	8.4	3.2	3.2	1.4	4.0	8.0	Q3
INA219AIDCNT	SOT-23	DCN	8	250	179.0	8.4	3.2	3.2	1.4	4.0	8.0	Q3
INA219BIDCNR	SOT-23	DCN	8	3000	179.0	8.4	3.2	3.2	1.4	4.0	8.0	Q3
INA219BIDCNT	SOT-23	DCN	8	250	179.0	8.4	3.2	3.2	1.4	4.0	8.0	Q3

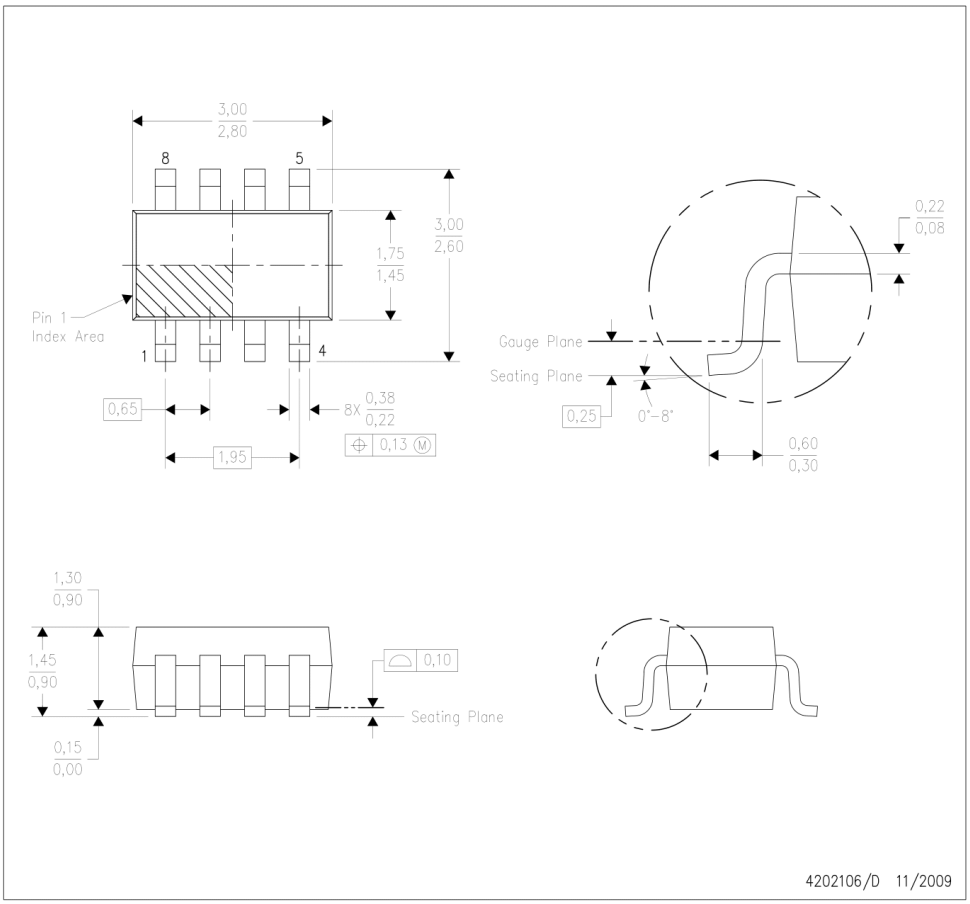
TAPE AND REEL BOX DIMENSIONS



*All dimensions are nominal

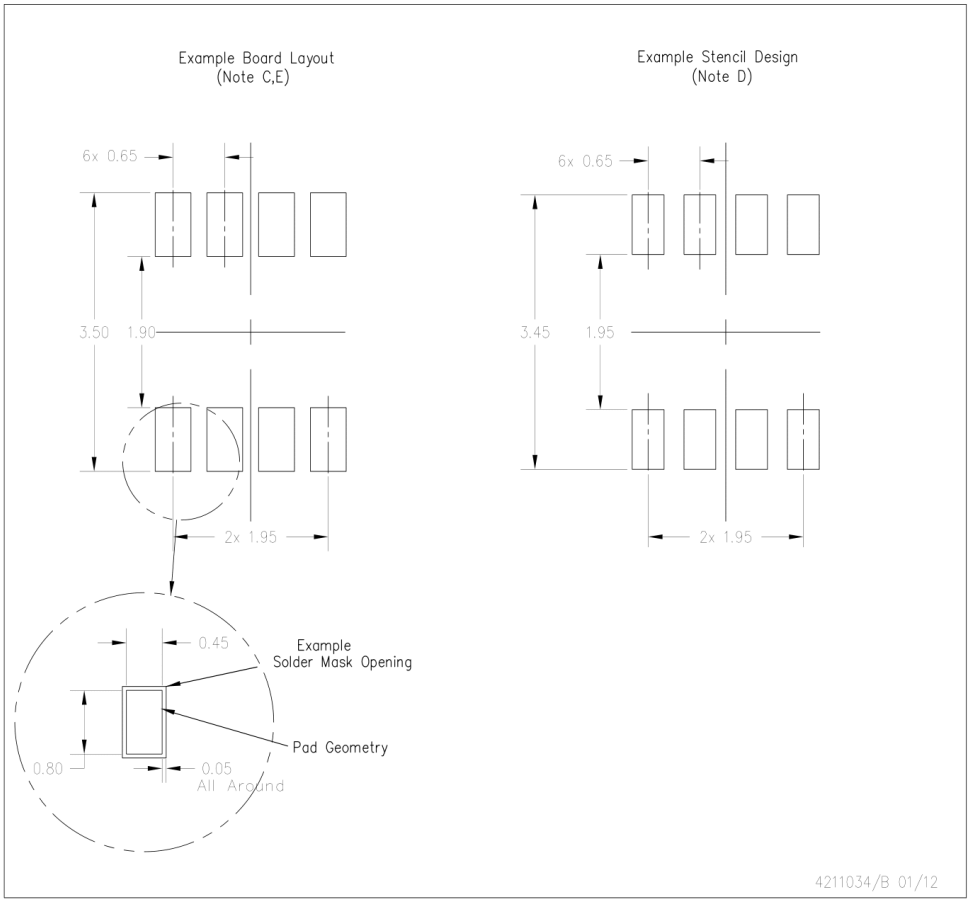
Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
INA219AIDCNR	SOT-23	DCN	8	3000	195.0	200.0	45.0
INA219AIDCNT	SOT-23	DCN	8	250	195.0	200.0	45.0
INA219BIDCNR	SOT-23	DCN	8	3000	195.0	200.0	45.0
INA219BIDCNT	SOT-23	DCN	8	250	195.0	200.0	45.0

DCN (R–PDSO–G8) PLASTIC SMALL–OUTLINE PACKAGE (DIE DOWN)



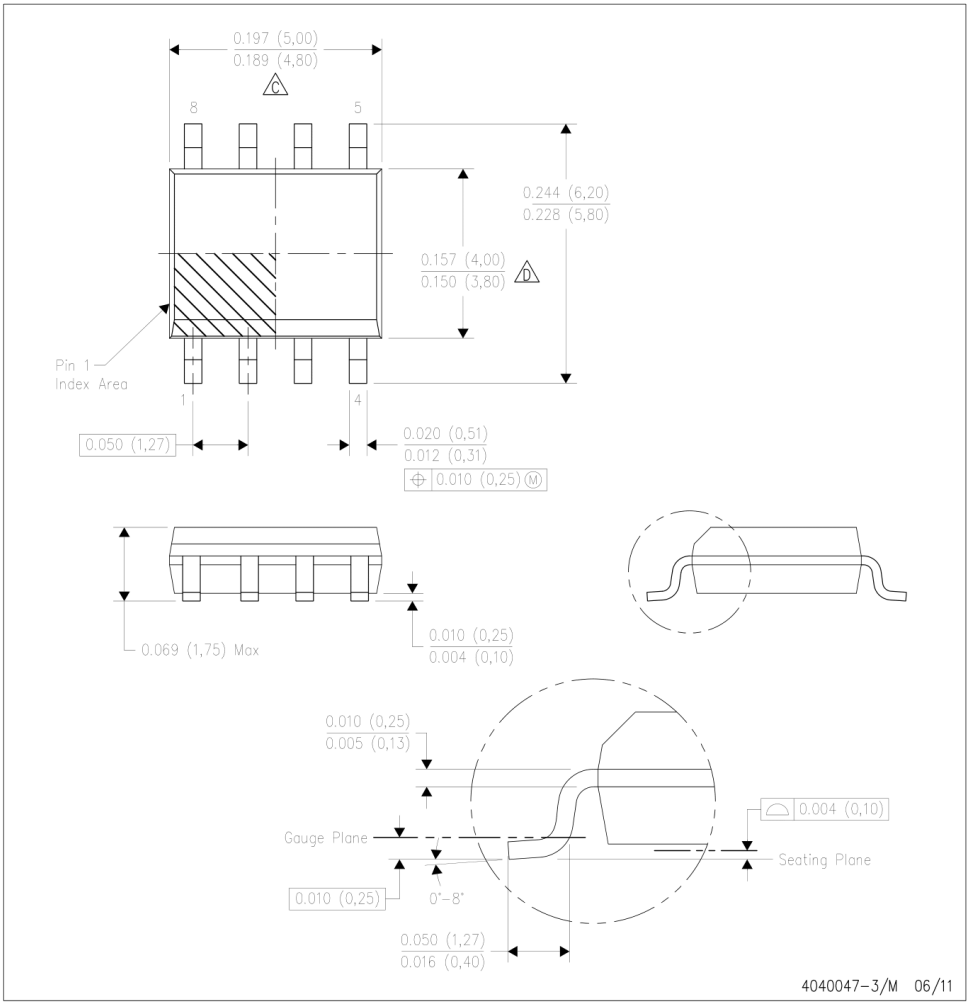
NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Package outline exclusive of metal burr & dambar protrusion/intrusion.
D. Package outline inclusive of solder plating.
E. A visual index feature must be located within the Pin 1 index area.
F. Falls within JEDEC MO–178 Variation BA.
G. Body dimensions do not include flash or protrusion. Mold flash and protrusion shall not exceed 0.25 per side.

DCN (R–PDSO–G8) PLASTIC SMALL–OUTLINE PACKAGE (DIE DOWN)



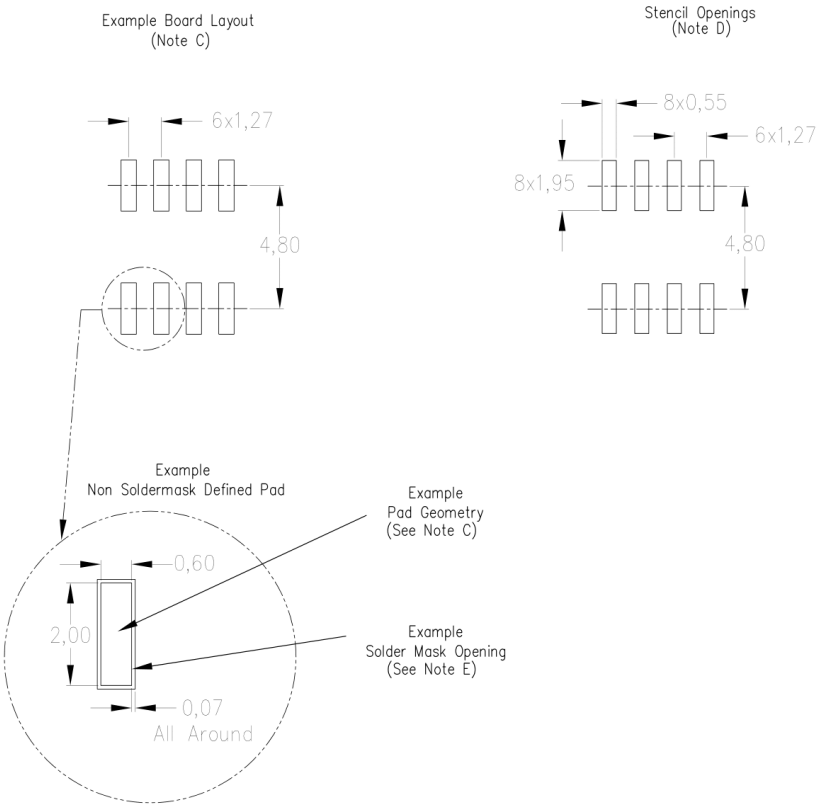
NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Publication IPC–7351 is recommended for alternate designs.
D. Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Refer to IPC–7525.
E. Customers should contact their board fabrication site for solder mask tolerances between and around signal pads.

D (R–PDSO–G8) PLASTIC SMALL OUTLINE



NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Body length does not include mold flash, protrusions, or gate burrs. Mold flash, protrusions, or gate burrs shall not exceed 0.006 (0,15) each side.
D. Body width does not include interlead flash. Interlead flash shall not exceed 0.017 (0,43) each side.
E. Reference JEDEC MS–012 variation AA.

D (R–PDSO–G8) PLASTIC SMALL OUTLINE



NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Publication IPC–7351 is recommended for alternate designs.
D. Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Refer to IPC–7525 for other stencil recommendations.
E. Customers should contact their board fabrication site for solder mask tolerances between and around signal pads.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Mobile Processors	www.ti.com/omap		
Wireless Connectivity	www.ti.com/wirelessconnectivity		
	TI E2E Community Home Page		e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Appendix G – INA226 Datasheet

INA226 36V, 16-Bit, Ultra-Precise I²C Output Current, Voltage, and Power Monitor With Alert

1 Features

- Senses bus voltages from 0V to 36V
- High-side or low-side sensing
- Reports current, voltage, and power
- High accuracy:
  - Gain error: 0.1% (maximum)
  - Offset: 10µV (maximum)
- Configurable averaging options
- 16 programmable addresses
- Operates from 2.7V to 5.5V power supply
- 10-pin DGS (VSSOP) package

2 Applications

- Rack servers
- Wireless infrastructure
- High performance computing
- Power management
- Battery cell monitors and balancers
- Power Supplies
- Test Equipment

3 Description

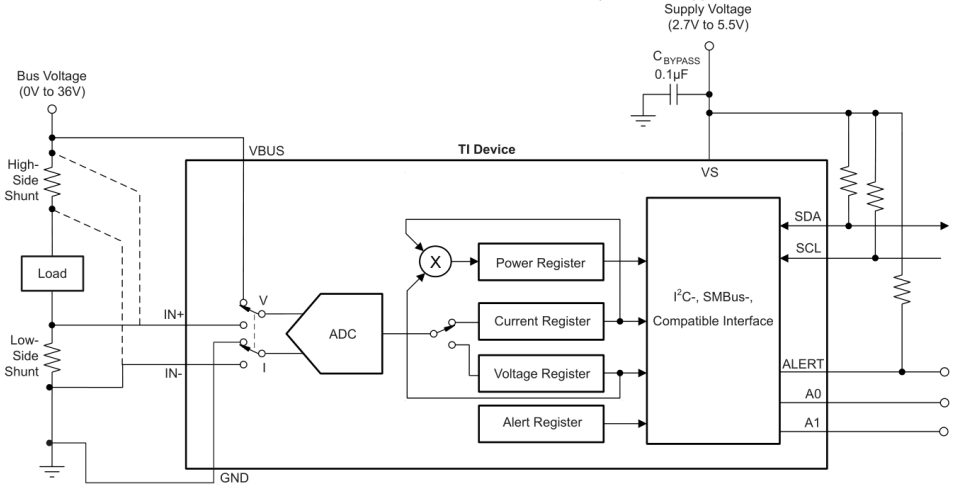
The INA226 is a current shunt and power monitor with an I2C™- or SMBUS-compatible interface. The device monitors both a shunt voltage drop and bus supply voltage. Programmable calibration value, conversion times, and averaging, combined with an internal multiplier, enable direct readouts of current in amperes and power in watts.

The INA226 senses current on common-mode bus voltages that can vary from 0V to 36V, independent of the supply voltage. The device operates from a single 2.7V to 5.5V supply, drawing a typical of 330µA of supply current. The device is specified over the operating temperature range between –40°C and 125°C and features up to 16 programmable addresses on the I²C-compatible interface.

Package Information

PART NUMBER	PACKAGE(1)	PACKAGE SIZE(2)
INA226	VSSOP (10)	3.00mm × 4.90mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.
(2) The package size (length × width) is a nominal value and includes pins, where applicable.



High-Side or Low-Side Sensing Application

Table of Contents

4 Pin Configuration and Functions

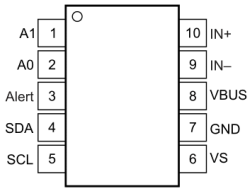


Figure 4-1. DGS Package 10-Pin VSSOP Top View

Table 4-1. Pin Functions

PIN		Type	DESCRIPTION
NAME	NO.		
A0	2	Digital input	Address pin. Connect to GND, SCL, SDA, or VS. Table 6-2 shows pin settings and corresponding addresses.
A1	1	Digital input	Address pin. Connect to GND, SCL, SDA, or VS. Table 6-2 shows pin settings and corresponding addresses.
Alert	3	Digital output	Multi-functional alert, open-drain output.
GND	7	Analog	Ground.
IN+	10	Analog input	Connect to supply side of shunt resistor.
IN–	9	Analog input	Connect to load side of shunt resistor.
SCL	5	Digital input	Serial bus clock line, open-drain input.
SDA	4	Digital I/O	Serial bus data line, open-drain input/output.
VBUS	8	Analog input	Bus voltage input.
VS	6	Analog	Power supply, 2.7 V to 5.5 V.

5 Specifications

5.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted) (1)

		MIN	MAX	UNIT
V_{VS}	Supply voltage		6	V
Analog Inputs, IN+, IN–	Differential (V_{IN+} – V_{IN-})(2)	–40	40	V
	Common-Mode (V_{IN+} + V_{IN-}) / 2	–0.3	40	
V_{VBUS}		–0.3	40	V
V_{SDA}		GND – 0.3	6	V
V_{SCL}		GND – 0.3	V_{VS} + 0.3	V
I_{IN}	Input current into any pin		5	mA
I_{OUT}	Open-drain digital output current		10	mA
T_J	Junction temperature		150	°C
T_{stg}	Storage temperature range	–65	150	°C

(1) Operation outside the *Absolute Maximum Ratings* may cause permanent device damage. *Absolute Maximum Ratings* do not imply functional operation of the device at these or any other conditions beyond those listed under *Recommended Operating Conditions*. If used outside the *Recommended Operating Conditions* but within the *Absolute Maximum Ratings*, the device may not be fully functional, and this may affect device reliability, functionality, performance, and shorten the device lifetime.
(2) IN+ and IN– can have a differential voltage between –40 V and 40 V. However, the voltage at these pins must not exceed the range –0.3 V to 40 V.

5.2 ESD Ratings

			VALUE	UNIT
$V_{(ESD)}$	Electrostatic discharge	Human body model (HBM), per ANSI/ESDA/JEDEC JS-001, all pins(1)	±2500	V
		Charged device model (CDM), per ANSI/ESDA/JEDEC JS-002, all pins(2)	±1000	

(1) JEDEC document JEP155 states that 500V HBM allows safe manufacturing with a standard ESD control process.
(2) JEDEC document JEP157 states that 250V CDM allows safe manufacturing with a standard ESD control process.

5.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

		MIN	NOM	MAX	UNIT
V_{CM}	Common-mode input voltage		12		V
V_{VS}	Operating supply voltage		3.3		V
T_A	Operating free-air temperature	–40		125	°C

5.4 Thermal Information

THERMAL METRIC(1)		INA226 DGS (VSSOP) 10 PINS	UNIT
$R_{\theta JA}$	Junction-to-ambient thermal resistance	144.6	°C/W
$R_{\theta JC(top)}$	Junction-to-case (top) thermal resistance	53.3	°C/W
$R_{\theta JB}$	Junction-to-board thermal resistance	80.4	°C/W
Ψ_{JT}	Junction-to-top characterization parameter	3.5	°C/W
Y_{JB}	Junction-to-board characterization parameter	78.9	°C/W
$R_{\theta JC(bot)}$	Junction-to-case (bottom) thermal resistance	N/A	°C/W

(1) For more information about traditional and new thermal metrics, see the Semiconductor and IC Package Thermal Metrics application note.

5.5 Electrical Characteristics

T_A = 25°C, V_{VS} = 3.3 V, V_{IN+} = 12 V, V_{SENSE} = (V_{IN+} – $V_{IN–}$) = 0 mV and V_{VBUS} = 12 V, unless otherwise noted

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
INPUT						
	Shunt voltage input range		–81.9175		81.92	mV
	Bus voltage input range(1)		0		36	V
CMRR	Common-mode rejection	0 V ≤ V_{IN+} ≤ 36 V	126	140		dB
V_{OS}	Shunt offset voltage, RTI(2)			±2.5	±10	µV
	Shunt offset voltage, RTI(2) vs temperature	–40°C ≤ T_A ≤ 125°C		0.02	0.1	µV/°C
PSRR	Shunt offset voltage, RTI(2) vs Power supply	2.7 V ≤ VS ≤ 5.5 V		±2.5		µV/V
V_{OS}	Bus offset voltage, RTI(2)			±1.25	±7.5	mV
	Bus offset voltage, RTI(2) vs temperature	–40°C ≤ T_A ≤ 125°C		10	40	µV/°C
PSRR	Bus offset voltage, RTI(2) vs power supply			±0.5		mV/V
I_B	Input bias current (I_{IN+}, $I_{IN–}$ pins)			0.1		nA
	VBUS input impedance			830		kΩ
	Input leakage (3)	(IN+ pin) + (IN– pin), Power-down mode		0.1	0.5	µA
DC ACCURACY						
	ADC native resolution			16		Bits
	1 LSB step size	Shunt voltage		2.5		µV
		Bus voltage		1.25		mV
	Shunt voltage gain error			0.02%	0.1%	
	Shunt voltage gain error vs temperature	–40°C ≤ T_A ≤ 125°C		10	50	ppm/°C
	Bus voltage gain error			0.02%	0.1%	
	Bus voltage gain error vs temperature	–40°C ≤ T_A ≤ 125°C		10	50	ppm/°C
	Differential nonlinearity			±0.1		LSB
t_{CT}	ADC conversion time	CT bit = 000		140	154	µs
		CT bit = 001		204	224	
		CT bit = 010		332	365	
		CT bit = 011		588	646	
		CT bit = 100		1.1	1.21	ms
		CT bit = 101		2.116	2.328	
		CT bit = 110		4.156	4.572	
		CT bit = 111		8.244	9.068	
SMBus						
	SMBus timeout(4)			28	35	ms
DIGITAL INPUT/OUTPUT						
	Input capacitance			3		pF
	Leakage input current	0 V ≤ V_{SCL} ≤ V_{VS} , 0 V ≤ V_{SDA} ≤ V_{VS}, 0 V ≤ V_{Alert} ≤ V_{VS} , 0 V ≤ V_{A0} ≤ V_{VS} , 0 V ≤ V_{A1} ≤ V_{VS}		0.1	1	µA
V_{IH}	High-level input voltage		0.7×V_{VS}		6	V
V_{IL}	Low-level input voltage		–0.5		0.3×V_{VS}	V
V_{OL}	Low-level output voltage, SDA, Alert	I_{OL} = 3 mA	0		0.4	V
	Hysteresis			500		mV
POWER SUPPLY						
	Operating supply range		2.7		5.5	V
I_Q	Quiescent current			330	420	µA
	Quiescent current, power-down (shutdown) mode			0.5	2	µA

T_A = 25°C, V_{VS} = 3.3 V, V_{IN+} = 12 V, V_{SENSE} = (V_{IN+} – $V_{IN–}$) = 0 mV and V_{VBUS} = 12 V, unless otherwise noted

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{POR}	Power-on reset threshold			2		V

(1) While the input range is 36 V, the full-scale range of the ADC scaling is 40.96 V. See the Section 6.3.1. Do not apply more than 36 V.
(2) RTI = Referred-to-input.
(3) Input leakage is positive (current flowing into the pin) for the conditions shown at the top of this table. Negative leakage currents can occur under different input conditions.
(4) SMBus timeout in the device resets the interface any time SCL is low for more than 28 ms.

5.6 Typical Characteristics

At T_A = 25°C, V_{VS} = 3.3 V, V_{IN+} = 12 V, V_{SENSE} = (V_{IN+} – $V_{IN–}$) = 0 mV and V_{VBUS} = 12 V, unless otherwise noted.

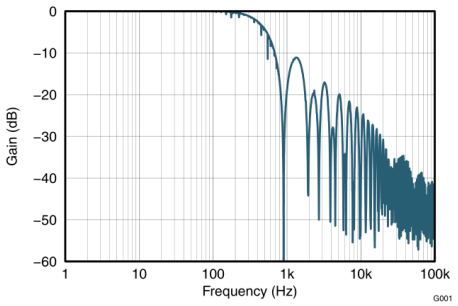


Figure 5-1. Frequency Response

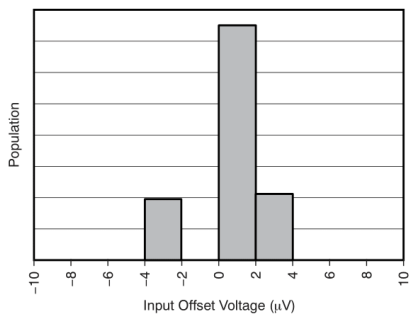


Figure 5-2. Shunt Input Offset Voltage Production Distribution

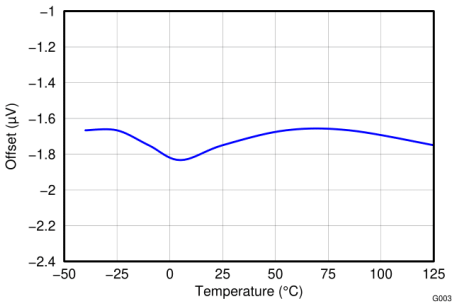


Figure 5-3. Shunt Input Offset Voltage vs Temperature

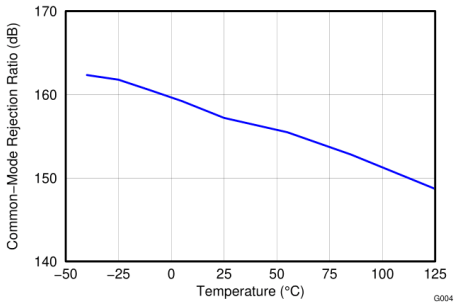


Figure 5-4. Shunt Input Common-Mode Rejection Ratio vs Temperature

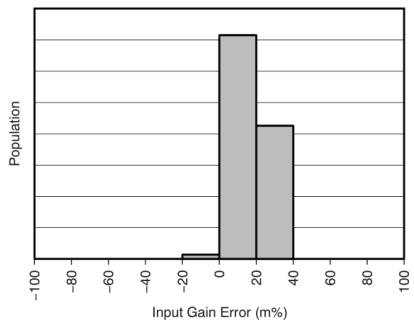


Figure 5-5. Shunt Input Gain Error Production Distribution

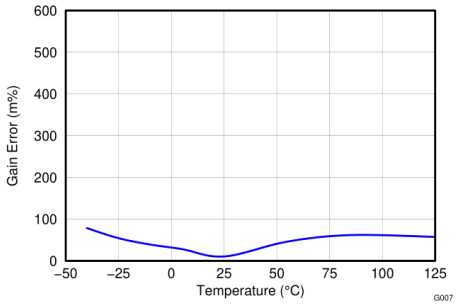


Figure 5-6. Shunt Input Gain Error vs Temperature

5.6 Typical Characteristics (continued)

At T_A = 25°C, V_{VS} = 3.3 V, V_{IN+} = 12 V, V_{SENSE} = (V_{IN+} – $V_{IN–}$) = 0 mV and V_{VBUS} = 12 V, unless otherwise noted.

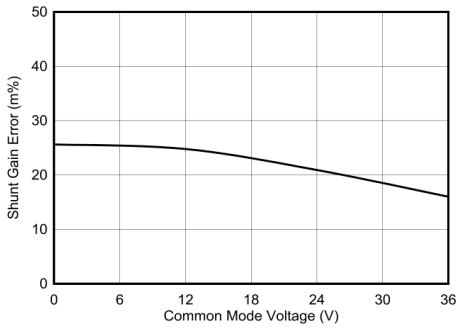


Figure 5-7. Shunt Input Gain Error vs Common-Mode Voltage

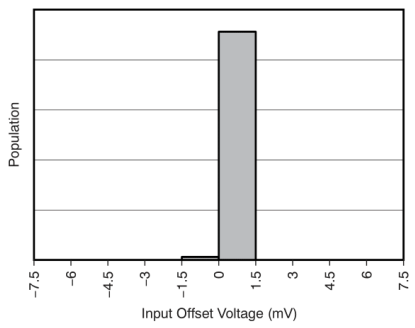


Figure 5-8. Bus Input Offset Voltage Production Distribution

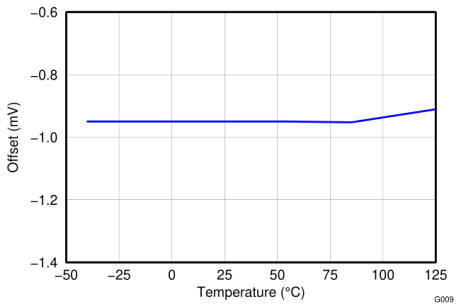


Figure 5-9. Bus Input Offset Voltage vs Temperature

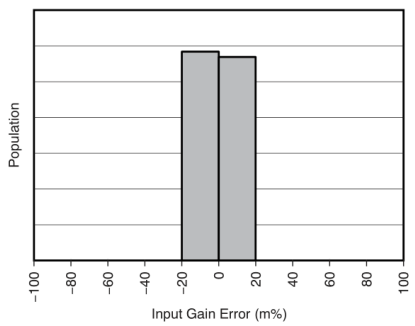


Figure 5-10. Bus Input Gain Error Production Distribution

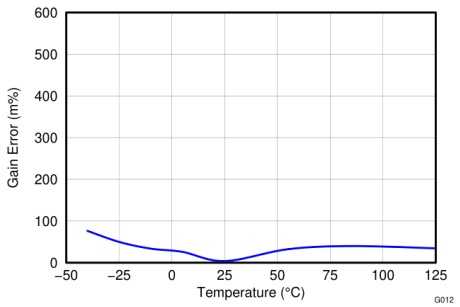


Figure 5-11. Bus Input Gain Error vs Temperature

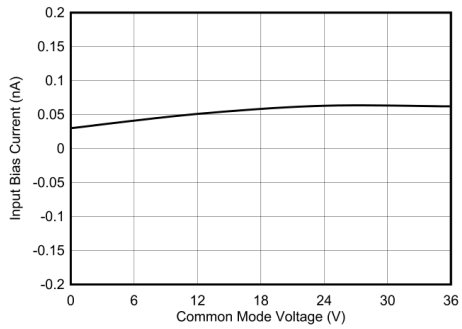


Figure 5-12. Input Bias Current vs Common-Mode Voltage

5.6 Typical Characteristics (continued)

At T_A = 25°C, V_{VS} = 3.3 V, V_{IN+} = 12 V, V_{SENSE} = (V_{IN+} – $V_{IN–}$) = 0 mV and V_{VBUS} = 12 V, unless otherwise noted.

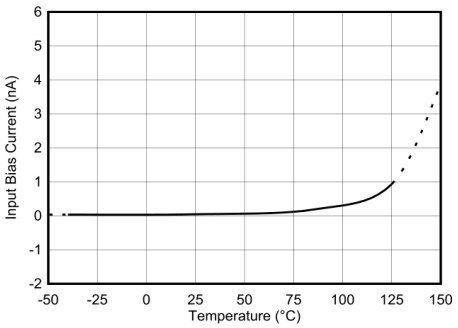


Figure 5-13. Input Bias Current vs Temperature

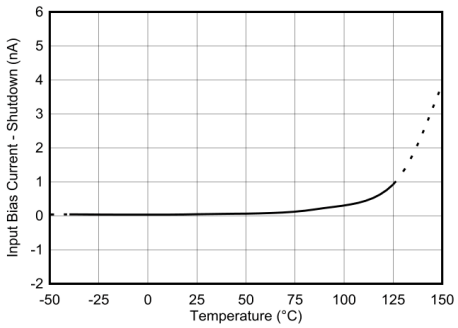


Figure 5-14. Input Bias Current vs Temperature, Shutdown

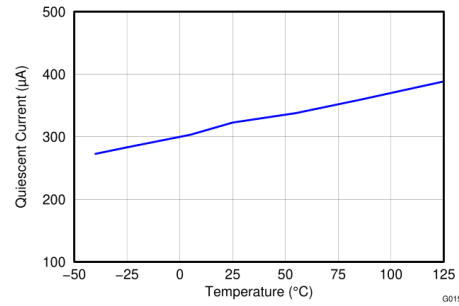


Figure 5-15. Active I_Q vs Temperature

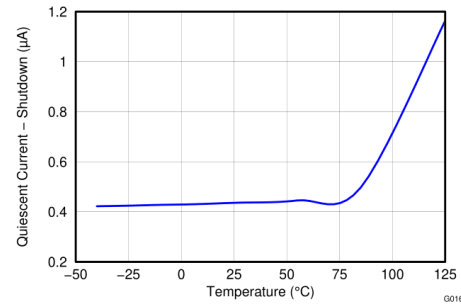


Figure 5-16. Shutdown I_Q vs Temperature

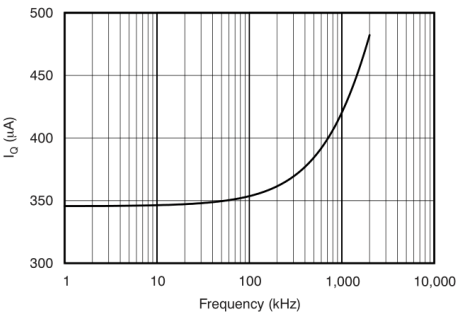


Figure 5-17. Active I_Q vs I^2C Clock Frequency

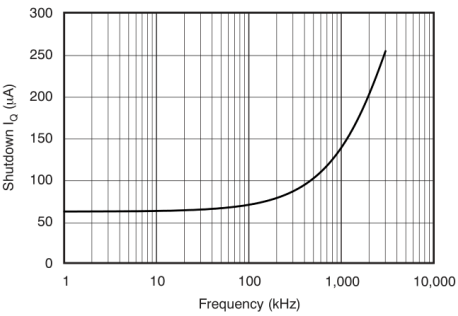


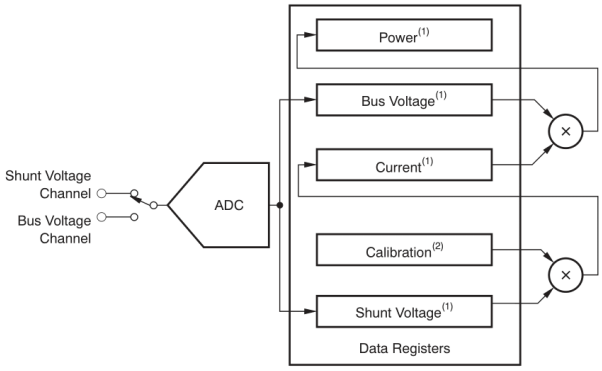
Figure 5-18. Shutdown I_Q vs I^2C Clock Frequency

6 Detailed Description

6.1 Overview

The INA226 is a digital current sense amplifier with an I²C- and SMBus-compatible interface. The device provides digital current, voltage, and power readings necessary for accurate decision-making in precisely-controlled systems. Programmable registers allow flexible configuration for measurement resolution as well as continuous-versus-triggered operation. Detailed register information appears at the end of this data sheet. See the *Functional Block Diagram* section for a block diagram of the INA226 device.

6.2 Functional Block Diagram



(1) Read-only
(2) Read/write

6.3 Feature Description

6.3.1 Basic ADC Functions

The INA226 device performs two measurements on the power-supply bus of interest. The voltage developed from the load current that flows through a shunt resistor creates a shunt voltage that is measured at the IN+ and IN– pins. The device can also measure the power supply bus voltage by connecting this voltage to the VBUS pin. The differential shunt voltage is measured with respect to the IN– pin while the bus voltage is measured with respect to ground.

The device is typically powered by a separate supply that can range from 2.7 V to 5.5 V. The bus that is being monitored can range in voltage from 0 V to 36 V. Based on the fixed 1.25-mV LSB for the Bus Voltage Register that a full-scale register results in a 40.96 V value.

Note

Do not apply more than 36 V of actual voltage to the input pins.

There are no special considerations for power-supply sequencing because the common-mode input range and power-supply voltage are independent of each other; therefore, the bus voltage can be present with the supply voltage off, and reciprocally.

The device takes two measurements, shunt voltage and bus voltage. The device then converts these measurements to current, based on the Calibration Register value, and then calculates power. Refer to the *Programming the Calibration Register* section for additional information on programming the Calibration Register.

The device has two operating modes, continuous and triggered, that determine how the ADC operates following these conversions. When the device is in the normal operating mode (that is, MODE bits of the Configuration Register (00h) are set to '111'), the device continuously converts a shunt voltage reading followed by a bus voltage reading. After the shunt voltage reading, the current value is calculated (based on Equation 3). This current value is then used to calculate the power result (using Equation 4). These values are subsequently stored in an accumulator, and the measurement/calculation sequence repeats until the number of averages set in the Configuration Register (00h) is reached. Following every sequence, the present set of values measured and calculated are appended to previously collected values. After all of the averaging has been completed, the final values for shunt voltage, bus voltage, current, and power are updated in the corresponding registers that can then be read. These values remain in the data output registers until the values are replaced by the next fully completed conversion results. Reading the data output registers does not affect a conversion in progress.

The mode control in the Conversion Register (00h) also permits selecting modes to convert only the shunt voltage or the bus voltage to further allow the user to configure the monitoring function to fit the specific application requirements.

All current and power calculations are performed in the background and do not contribute to conversion time.

In triggered mode, writing any of the triggered convert modes into the Configuration Register (00h) (that is, MODE bits of the Configuration Register (00h) are set to ‘001’, ‘010’, or ‘011’) triggers a single-shot conversion. This action produces a single set of measurements; thus, to trigger another single-shot conversion, the Configuration Register (00h) must be written to a second time, even if the mode does not change.

In addition to the two operating modes (continuous and triggered), the device also has a power-down mode that reduces the quiescent current and turns off current into the device inputs, reducing the impact of supply drain when the device is not being used. Full recovery from power-down mode requires 40µs. The registers of the device can be written to and read from while the device is in power-down mode. The device remains in power-down mode until one of the active modes settings are written into the Configuration Register (00h).

Although the device can be read at any time, and the data from the last conversion remain available, the Conversion Ready flag bit (Mask/Enable Register, CVRF bit) is provided to help coordinate one-shot or triggered conversions. The Conversion Ready flag (CVRF) bit is set after all conversions, averaging, and multiplication operations are complete.

The Conversion Ready flag (CVRF) bit clears under these conditions:

- Writing to the Configuration Register (00h), except when configuring the MODE bits for power-down mode; or
- Reading the Mask/Enable Register (06h)

6.3.1.1 Power Calculation

The Current and Power are calculated following shunt voltage and bus voltage measurements as shown in Figure 6-1. Current is calculated following a shunt voltage measurement based on the value set in the Calibration Register. If there is no value loaded into the Calibration Register, the current value stored is zero. Power is calculated following the bus voltage measurement based on the previous current calculation and bus voltage measurement. If there is no value loaded in the Calibration Register, the power value stored is also zero. Again, these calculations are performed in the background and do not add to the overall conversion time. These current and power values are considered intermediate results (unless the averaging is set to 1) and are stored in an internal accumulation register, not the corresponding output registers. Following every measured sample, the newly-calculated values for current and power are appended to this accumulation register until all of the samples have been measured and averaged based on the number of averages set in the Configuration Register (00h).

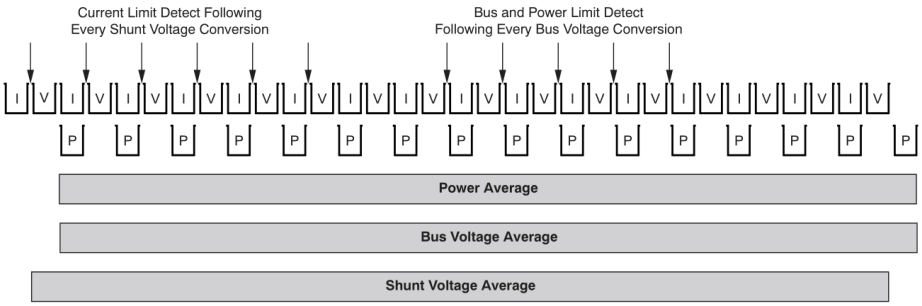


Figure 6-1. Power Calculation Scheme

In addition to the current and power accumulating after every sample, the shunt and bus voltage measurements are also collected. After all of the samples have been measured and the corresponding current and power calculations have been made, the accumulated average for each of these parameters is then loaded to the corresponding output registers, where the average can then be read.

6.3.1.2 Alert Pin

The INA226 has a single Alert Limit Register (07h), that allows the Alert pin to be programmed to respond to a single user-defined event or to a Conversion Ready notification if desired. The Mask/Enable Register allows the user to select from one of the five available functions to monitor and/or set the Conversion Ready bit to control the response of the Alert pin. Based on the function being monitored, the user then enters a value into the Alert Limit Register to set the corresponding threshold value that asserts the Alert pin.

The Alert pin allows for one of several available alert functions to be monitored to determine if a user-defined threshold has been exceeded. The five alert functions that can be monitored are:

- Shunt Voltage Over-Limit (SOL)
- Shunt Voltage Under-Limit (SUL)
- Bus Voltage Over-Limit (BOL)
- Bus Voltage Under-Limit (BUL)
- Power Over-Limit (POL)

The Alert pin is an open-drain output. This pin is asserted when the alert function selected in the Mask/Enable Register exceeds the value programmed into the Alert Limit Register. Only one of these alert functions can be enabled and monitored at a time. If multiple alert functions are enabled, the selected function in the highest significant bit position takes priority and responds to the Alert Limit Register value. For example, if the Shunt Voltage Over-Limit function and the Shunt Voltage Under-Limit function are both selected, the Alert pin asserts when the Shunt Voltage Register exceeds the value in the Alert Limit Register.

The Conversion Ready state of the device can also be monitored at the Alert pin to inform the user when the device has completed the previous conversion and is ready to begin a new conversion. Conversion Ready can be monitored at the Alert pin along with one of the alert functions. If an alert function and the Conversion Ready are both enabled to be monitored at the Alert pin, after the Alert pin is asserted, the Mask/Enable Register must be read following the alert to determine the source of the alert. By reading the Conversion Ready Flag (CVRF, bit 3), and the Alert Function Flag (AFF, bit 4) in the Mask/Enable Register, the source of the alert can be determined. If the Conversion Ready feature is not desired and the CNVR bit is not set, the Alert pin only responds to an exceeded alert limit based on the alert function enabled.

If the alert function is not used, the Alert pin can be left floating without impacting the operation of the device.

Refer to Figure 6-1 to see the relative timing of when the value in the Alert Limit Register is compared to the corresponding converted value. For example, if the alert function that is enabled is Shunt Voltage Over-Limit (SOL), following every shunt voltage conversion the value in the Alert Limit Register is compared to the

measured shunt voltage to determine if the measurements has exceeded the programmed limit. The AFF, bit 4 of the Mask/Enable Register, asserts high any time the measured voltage exceeds the value programmed into the Alert Limit Register. In addition to the AFF being asserted, the Alert pin is asserted based on the Alert Polarity Bit (APOL, bit 1 of the Mask/Enable Register). If the Alert Latch is enabled, the AFF and Alert pin remain asserted until either the Configuration Register (00h) is written to or the Mask/Enable Register is read.

The Bus Voltage alert functions compare the measured bus voltage to the Alert Limit Register following every bus voltage conversion and assert the AFF bit and Alert pin if the limit threshold is exceeded.

The Power Over-Limit alert function is also compared to the calculated power value following every bus voltage measurement conversion and asserts the AFF bit and Alert pin if the limit threshold is exceeded.

6.4 Device Functional Modes

6.4.1 Averaging and Conversion Time Considerations

The INA226 device offers programmable conversion times (t_{CT}) for both the shunt voltage and bus voltage measurements. The conversion times for these measurements can be selected from as fast as 140 µs to as long as 8.244 ms. The conversion time settings, along with the programmable averaging mode, allow the device to be configured to optimize the available timing requirements in a given application. For example, if a system requires that data be read every 5ms, the device can be configured with the conversion times set to 588 µs for both shunt and bus voltage measurements and the averaging mode set to 4. This configuration results in the data updating approximately every 4.7ms. The device can also be configured with a different conversion time setting for the shunt and bus voltage measurements. This type of approach is common in applications where the bus voltage tends to be relatively stable. This situation can allow for the time focused on the bus voltage measurement to be reduced relative to the shunt voltage measurement. The shunt voltage conversion time can be set to 4.156 ms with the bus voltage conversion time set to 588 µs, with the averaging mode set to 1. This configuration also results in data updating approximately every 4.7 ms.

There are trade-offs associated with the settings for conversion time and the averaging mode used. The averaging feature can significantly improve the measurement accuracy by effectively filtering the signal. This approach allows the device to reduce any noise in the measurement that can be caused by noise coupling into the signal. A greater number of averages enables the device to be more effective in reducing the noise component of the measurement.

The conversion times selected can also have an impact on the measurement accuracy. Figure 6-2 shows multiple conversion times to illustrate the impact of noise on the measurement. to achieve the highest accuracy measurement possible, use a combination of the longest allowable conversion times and highest number of averages, based on the timing requirements of the system.

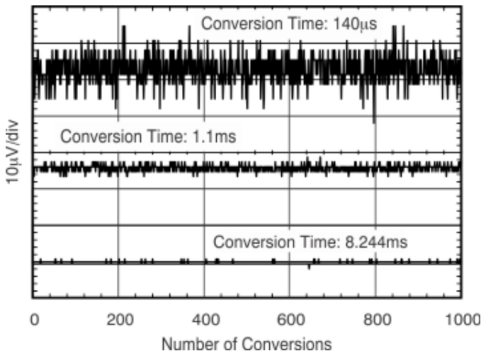


Figure 6-2. Noise vs Conversion Time

6.4.2 Filtering and Input Considerations

Measuring current is often noisy, and such noise can be difficult to define. The INA226 device offers several options for filtering by allowing the conversion times and number of averages to be selected independently in the Configuration Register (00h). The conversion times can be set independently for the shunt voltage and bus voltage measurements to allow added flexibility in configuring the monitoring of the power-supply bus.

The internal ADC is based on a delta-sigma (ΔΣ) front-end with a 500 kHz (±30%) typical sampling rate. This architecture has good inherent noise rejection; however, transients that occur at or very close to the sampling rate harmonics can cause problems. Because these signals are at 1 MHz and higher, the signals can be managed by incorporating filtering at the input of the device. The high frequency enables the use of low-value series resistors on the filter with negligible effects on measurement accuracy. In general, filtering the device input is only necessary if there are transients at exact harmonics of the 500 kHz (±30%) sampling rate (greater than 1 MHz). Filter using the lowest possible series resistance (typically 10 Ω or less) and a ceramic capacitor. Recommended values for this capacitor are between 0.1 µF and 1 µF. Figure 6-3 shows the device with a filter added at the input.

Overload conditions are another consideration for the device inputs. The device inputs are specified to tolerate 40 V across the inputs. A large differential scenario can be a short to ground on the load side of the shunt. This type of event can result in full power-supply voltage across the shunt (as long the power supply or energy storage capacitors support the voltage). Removing a short to ground can result in inductive kickbacks that can exceed the 40-V differential and common-mode rating of the device. Inductive kickback voltages are best controlled by Zener-type transient-absorbing devices (commonly called *transzorbs*) combined with sufficient energy storage capacitance. See the TI Design, *Transient Robustness for Current Shunt Monitors* (TIDU473), which describes a high-side current shunt monitor used to measure the voltage developed across a current-sensing resistor when current passes through the resistor.

In applications that do not have large energy storage electrolytics on one or both sides of the shunt, an input overstress condition can result from an excessive dV/dt of the voltage applied to the input. A hard physical short is the most likely cause of this event, particularly in applications with no large electrolytics present. This problem occurs because an excessive dV/dt can activate the ESD protection in the device in systems where large currents are available. Testing demonstrates that the addition of 10-Ω resistors in series with each input of the device sufficiently protects the inputs against this dV/dt failure up to the 40-V rating of the device. Selecting these resistors in the range noted has minimal effect on accuracy.

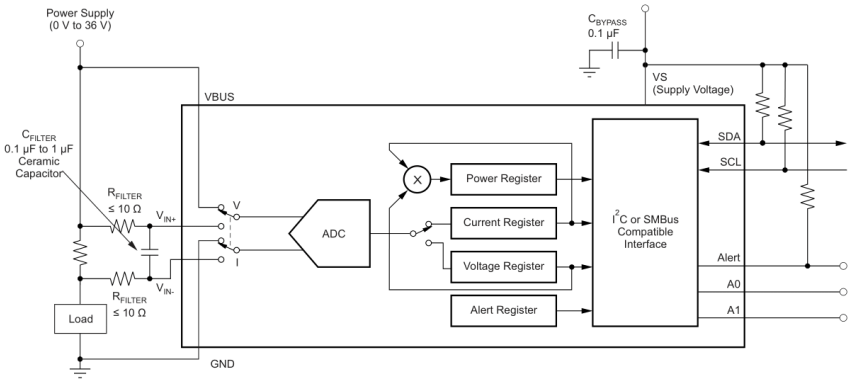


Figure 6-3. Input Filtering

6.5 Programming

An important aspect of the INA226 is that the device does not necessarily measure current or power. The device measures both the differential voltage applied between the IN+ and IN- input pins and the voltage applied to

the VBUS pin. For the device to report both current and power values, the user must program the resolution of the Current Register (04h) and the value of the shunt resistor present in the application to develop the differential voltage applied between the input pins. The Power Register (03h) is internally set to be 25 times the programmed Current_LSB. Both the Current_LSB and shunt resistor value are used in the calculation of the Calibration Register value the device uses to calculate the corresponding current and power values based on the measured shunt and bus voltages.

The Calibration Register is calculated based on Equation 1. This equation includes the term Current_LSB, which is the programmed value for the LSB for the Current Register (04h). The user uses this value to convert the value in the Current Register (04h) to the actual current in amperes. The highest resolution for the Current Register (04h) can be obtained by using the smallest allowable Current_LSB based on the maximum expected current as shown in Equation 2. While this value yields the highest resolution, selecting a value for the Current_LSB to the nearest round number above this value is common to simplify the conversion of the Current Register (04h) and Power Register (03h) to amperes and watts respectively. The R_{SHUNT} term is the value of the external shunt used to develop the differential voltage across the input pins.

$$\text{CAL} = \frac{0.00512}{\text{Current\_LSB} \times R_{SHUNT}} \tag{1}$$

where

- 0.00512 is an internal fixed value used to verify scaling is maintained properly

$$\text{Current\_LSB} = \frac{\text{Maximum Expected Current}}{2^{15}} \tag{2}$$

After programming the Calibration Register, the Current Register (04h) and Power Register (03h) update accordingly based on the corresponding shunt voltage and bus voltage measurements. Until the Calibration Register is programmed, the Current Register (04h) and Power Register (03h) remain at zero.

6.5.1 Programming the Calibration Register

Figure 8-1 shows a nominal 10-A load that creates a differential voltage of 20 mV across a 2-mΩ shunt resistor. The bus voltage for the INA226 is measured at the external VBUS input pin, which in this example is connected to the IN– pin to measure the voltage level delivered to the load. For this example, the VBUS pin measures less than 12 V because the voltage at the IN– pin is 11.98 V as a result of the voltage drop across the shunt resistor.

For this example, assuming a maximum expected current of 15 A, the Current_LSB is calculated to be 457.7 µA/bit using Equation 2. Using a value for the Current_LSB of 500 µA/Bit or 1 mA/Bit significantly simplifies the conversion from the Current Register (04h) and Power Register (03h) to amperes and watts. For this example, a value of 1 mA/bit is selected for the Current_LSB. Using this value for the Current_LSB does trade a small amount of resolution for having a simpler conversion process on the user side. Using Equation 1 in this example with a Current_LSB value of 1 mA/bit and a shunt resistor of 2 mΩ results in a Calibration Register value of 2560, or A00h.

The Current Register (04h) is then calculated by multiplying the decimal value of the Shunt Voltage Register (01h) contents by the decimal value of the Calibration Register and then dividing by 2048, as shown in Equation 3. For this example, the Shunt Voltage Register contains a value of 8,000 (representing 20 mV), which is multiplied by the Calibration Register value of 2560 and then divided by 2048 to yield a decimal value for the Current Register (04h) of 10000, or 2710h. Multiplying this value by 1 mA/bit results in the original 10-A level stated in the example.

$$\text{Current} = \frac{\text{ShuntVoltage} \times \text{CalibrationRegister}}{2048} \tag{3}$$

The LSB for the Bus Voltage Register (02h) is a fixed 1.25 mV/bit, which means that the 11.98 V present at the VBUS pin results in a register value of 2570h, or a decimal equivalent of 9584. Note that the MSB of the Bus Voltage Register (02h) is always zero because the VBUS pin is only able to measure positive voltages.

The Power Register (03h) is then be calculated by multiplying the decimal value of the Current Register, 10000, by the decimal value of the Bus Voltage Register (02h), 9584, and then dividing by 20,000, as defined in Equation 4. For this example, the result for the Power Register (03h) is 12B8h, or a decimal equivalent of 4792. Multiplying this result by the power LSB (25 times the [1 × 10^{-3} Current_LSB]) results in a power calculation of (4792 × 25 mW/bit), or 119.82 W. The power LSB has a fixed ratio to the Current_LSB of 25. For this example, a programmed 1 mA/bit Current_LSB results in a power LSB of 25 mW/bit. This ratio is internally programmed to verify that the scaling of the power calculation is within an acceptable range. A manual calculation for the power being delivered to the load uses a bus voltage of 11.98 V (12 V_{CM} – 20 mV shunt drop) multiplied by the load current of 10 A to give a result of 119.8 W.

$$\text{Power} = \frac{\text{Current} \times \text{BusVoltage}}{20000} \tag{4}$$

Table 6-1 lists the steps for configuring, measuring, and calculating the values for current and power for this device.

Table 6-1. Calculating Current and Power

STEP	REGISTER NAME	ADDRESS	CONTENTS	DEC	LSB	VALUE (1)
Step 1	Configuration Register	00h	4127h	—	—	—
Step 2	Shunt Register	01h	1F40h	8000	2.5 µV	20 mV
Step 3	Bus Voltage Register	02h	2570h	9584	1.25 mV	11.98 V
Step 4	Calibration Register	05h	A00h	2560	—	—
Step 5	Current Register	04h	2710	10000	1 mA	10 A
Step 6	Power Register	03h	12B8h	4792	25 mW	119.82 W

(1) Conditions: Load = 10 A, V_{CM} = 12 V, R_{SHUNT} = 2 mΩ, and V_{VBUS} = 12 V.

6.5.2 Programming the Power Measurement Engine

6.5.2.1 Calibration Register and Scaling

The Calibration Register enables the user to scale the Current Register (04h) and Power Register (03h) to the most useful value for a given application. For example, set the Calibration Register such that the largest possible number is generated in the Current Register (04h) or Power Register (03h) at the expected full-scale point. This approach yields the highest resolution using the previously calculated minimum Current_LSB in the equation for the Calibration Register. The Calibration Register can also be selected to provide values in the Current Register (04h) and Power Register (03h) that either provide direct decimal equivalents of the values being measured, or yield a round LSB value for each corresponding register. After these choices are made, the Calibration Register also offers possibilities for end user system-level calibration. After determining the exact current by using an external ammeter, the value of the Calibration Register can then be adjusted based on the measured current result of the INA226 to cancel the total system error as shown in Equation 5.

$$\text{Corrected\_Full\_Scale\_Cal} = \text{trunc}\left[\frac{\text{Cal} \times \text{MeasShuntCurrent}}{\text{Device\_Current}}\right] \tag{5}$$

6.5.3 Simple Current Shunt Monitor Usage (No Programming Necessary)

The device can be used without any programming if reading a shunt voltage drop and bus voltage with the default power-on reset configuration and continuous conversion of shunt and bus voltages is only necessary.

Without programming the device Calibration Register, the device is unable to provide either a valid current or power value, because these outputs are both derived using the values loaded into the Calibration Register.

6.5.4 Default Settings

The default power-up states of the registers are shown in the *Register Maps* section of this data sheet. These registers are volatile, and if programmed to a value other than the default values shown in Table 7-1, the registers must be re-programmed at every device power-up. Detailed information on programming the Calibration Register specifically is given in the *Programming* section and calculated based on Equation 1.

6.5.5 Bus Overview

The INA226 offers compatibility with both I^{2}C and SMBus interfaces. The I^{2}C and SMBus protocols are essentially compatible with one another.

The I^{2}C interface is used throughout this data sheet as the primary example, with SMBus protocol specified only when a difference between the two systems is discussed. Two lines, SCL and SDA, connect the device to the bus. Both SCL and SDA are open-drain connections.

The device that initiates a data transfer is called a *master*, and the devices controlled by the master are *slaves*. The bus must be controlled by a master device that generates the serial clock (SCL), controls the bus access, and generates START and STOP conditions.

To address a specific device, the master initiates a start condition by pulling the data signal line (SDA) from a high to a low logic level while SCL is high. All slaves on the bus shift in the slave address byte on the rising edge of SCL, with the last bit indicating whether a read or write operation is intended. During the ninth clock pulse, the slave being addressed responds to the master by generating an Acknowledge and pulling SDA low.

Data transfer is then initiated and eight bits of data are sent, followed by an *Acknowledge* bit. During data transfer, SDA must remain stable while SCL is high. Any change in SDA while SCL is high is interpreted as a start or stop condition.

After all data have been transferred, the master generates a stop condition, indicated by pulling SDA from low to high while SCL is high. The device includes a 28 ms timeout on the interface to prevent locking up the bus.

6.5.5.1 Serial Bus Address

To communicate with the INA226, the master must first address slave devices via a slave address byte. The slave address byte consists of seven address bits and a direction bit that indicates whether the action is to be a read or write operation.

The device has two address pins, A0 and A1. Table 6-2 lists the pin logic levels for each of the 16 possible addresses. The device samples the state of pins A0 and A1 on every bus communication. Establish the pin states before any activity on the interface occurs.

Table 6-2. Address Pins and Slave Addresses

A1	A0	SLAVE ADDRESS
GND	GND	1000000
GND	VS	1000001
GND	SDA	1000010
GND	SCL	1000011
VS	GND	1000100
VS	VS	1000101
VS	SDA	1000110
VS	SCL	1000111
SDA	GND	1001000
SDA	VS	1001001
SDA	SDA	1001010
SDA	SCL	1001011
SCL	GND	1001100
SCL	VS	1001101
SCL	SDA	1001110
SCL	SCL	1001111

6.5.5.2 Serial Interface

The INA226 operates only as a slave device on both the I^2C bus and the SMBus. Connections to the bus are made using the open-drain SDA and SCL lines. The SDA and SCL pins feature integrated spike suppression filters and Schmitt triggers to minimize the effects of input spikes and bus noise. Although the device integrates spike suppression into the digital I/O lines, proper layout techniques help minimize the amount of coupling into the communication lines. This noise introduction can occur from capacitively coupling signal edges between the two communication lines themselves or from other switching noise sources present in the system. Routing traces in parallel with ground in between layers on a printed circuit board (PCB) typically reduces the effects of coupling between the communication lines. Shielded communication lines reduces the possibility of unintended noise coupling into the digital I/O lines that can be incorrectly interpreted as start or stop commands.

The INA226 supports the transmission protocol for fast mode (1 kHz to 400 kHz) and high-speed mode (1 kHz to 2.94 MHz). All data bytes are transmitted most significant byte first.

6.5.5.3 Writing to and Reading From the INA226

Accessing a specific register on the INA226 is accomplished by writing the appropriate value to the register pointer. Refer to Table 7-1 for a complete list of registers and corresponding addresses. The value for the register pointer (as shown in Figure 6-7) is the first byte transferred after the slave address byte with the R/ $\overline{W}$ bit low. Every write operation to the device requires a value for the register pointer.

Writing to a register begins with the first byte transmitted by the master. This byte is the slave address, with the R/ $\overline{W}$ bit low. The device then acknowledges receipt of a valid address. The next byte transmitted by the master is the address of the register which data is written to. This register address value updates the register pointer to the desired register. The next two bytes are written to the register addressed by the register pointer. The device

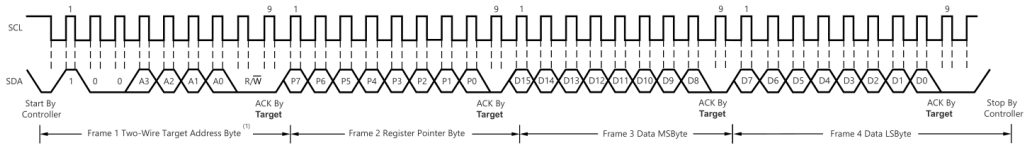
acknowledges receipt of each data byte. The master can terminate data transfer by generating a start or stop condition.

When reading from the device, the last value stored in the register pointer by a write operation determines which register is read during a read operation. To change the register pointer for a read operation, a new value must be written to the register pointer. This write is accomplished by issuing a slave address byte with the R/ $\overline{W}$ bit low, followed by the register pointer byte. No additional data are required. The master then generates a start condition and sends the slave address byte with the R/ $\overline{W}$ bit high to initiate the read command. The next byte is transmitted by the slave and is the most significant byte of the register indicated by the register pointer. This byte is followed by an *Acknowledge* from the master; then the slave transmits the least significant byte. The master acknowledges receipt of the data byte. The master can terminate data transfer by generating a *Not-Acknowledge* after receiving any data byte, or generating a start or stop condition. If repeated reads from the same register are desired, continually sending the register pointer bytes is not necessary; the device retains the register pointer value until the value is changed by the next write operation.

Figure 6-4 shows the write operation timing diagram. Figure 6-5 shows the read operation timing diagram.

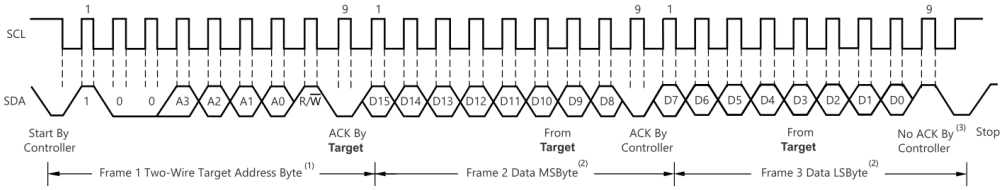
Note

Register bytes are sent most-significant byte first, followed by the least significant byte.



1. The value of the Slave Address byte is determined by the settings of the A0 and A1 pins. Refer to Table 6-2.

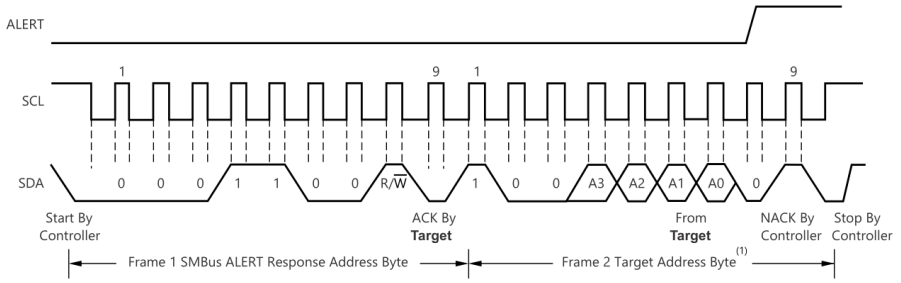
Figure 6-4. Timing Diagram for Write Word Format



1. The value of the Slave Address byte is determined by the settings of the A0 and A1 pins. Refer to Table 6-2.
2. Read data is from the last register pointer location. If a new register is desired, the register pointer must be updated. See Figure 6-7.
3. ACK by Master can also be sent.

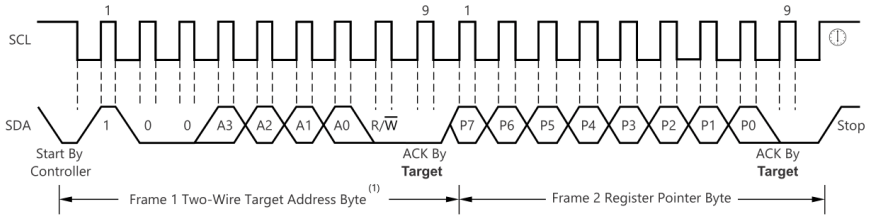
Figure 6-5. Timing Diagram for Read Word Format

Figure 6-6 shows the timing diagram for the SMBus Alert response operation. Figure 6-7 illustrates a typical register pointer configuration.



1. The value of the Slave Address Byte is determined by the settings of the A0 and A1 pins. Refer to Table 6-2.

Figure 6-6. Timing Diagram for SMBus ALERT



1. The value of the Slave Address Byte is determined by the settings of the A0 and A1 pins. Refer to Table 6-2.

Figure 6-7. Typical Register Pointer Set

6.5.5.3.1 High-Speed I²C Mode

When the bus is idle, both the SDA and SCL lines are pulled high by the pullup resistors. The master generates a start condition followed by a valid serial byte containing high-speed (HS) master code *00001XXX*. This transmission is made in fast (400 kHz) or standard (100 kHz) (F/S) mode at no more than 400 kHz. The device does not acknowledge the HS master code, but does recognize the code and switches the internal filters to support 2.94 MHz operation.

The master then generates a repeated start condition (a repeated start condition has the same timing as the start condition). After this repeated start condition, the protocol is the same as F/S mode, except that transmission speeds up to 2.94 MHz are allowed. Instead of using a stop condition, use repeated start conditions to secure the bus in HS-mode. A stop condition ends the HS-mode and switches all the internal filters of the device to support the F/S mode.

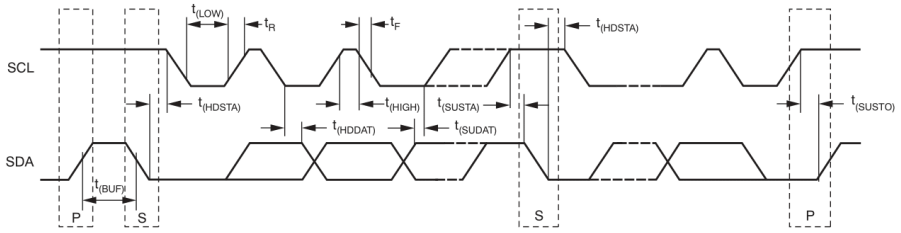


Figure 6-8. Bus Timing Diagram

Table 6-3. Bus Timing Diagram Definitions (1)

PARAMETER		FAST MODE		HIGH-SPEED MODE		UNIT
		MIN	MAX	MIN	MAX	
SCL operating frequency	$f_{(SCL)}$	0.001	0.4	0.001	2.94	MHz
Bus free time between stop and start conditions	$t_{(BUF)}$	600		160		ns
Hold time after repeated START condition. After this period, the first clock is generated.	$t_{(HDSTA)}$	100		100		ns
Repeated start condition setup time	$t_{(SUSTA)}$	100		100		ns
STOP condition setup time	$t_{(SUSTO)}$	100		100		ns
Data hold time	$t_{(HDDAT)}$	10	900	10	100	ns
Data setup time	$t_{(SUDAT)}$	100		20		ns
SCL clock low period	$t_{(LOW)}$	1300		200		ns
SCL clock high period	$t_{(HIGH)}$	600		60		ns
Data fall time	t_F		300		80	ns
Clock fall time	t_F		300		40	ns
Clock rise time	t_R		300		40	ns
Clock/data rise time for SCLK ≤ 100kHz	t_R		1000			ns

(1) Values based on a statistical analysis of a one-time sample of devices. Minimum and maximum values are not guaranteed and not production tested.

6.5.5.4 SMBus Alert Response

The INA226 is designed to respond to the SMBus Alert Response address. The SMBus Alert Response provides a quick fault identification for simple slave devices. When an Alert occurs, the master can broadcast the Alert Response slave address (0001 100) with the Read/Write bit set high. Following this Alert Response, any slave device that generates an alert identifies the slave device by acknowledging the Alert Response and sending the address on the bus.

The Alert Response can activate several different slave devices simultaneously, similar to the I^2C General Call. If more than one slave attempts to respond, bus arbitration rules apply. The losing device does not generate an Acknowledge and continues to hold the Alert line low until the interrupt is cleared.

7 Registers

7.1 Register Maps

The INA226 uses a bank of registers for holding configuration settings, measurement results, minimum/maximum limits, and status information. Table 7-1 summarizes the device registers; refer to the *Functional Block Diagram* section for an illustration of the registers.

All 16-bit device registers are two 8-bit bytes via the I^2C interface.

Table 7-1. Register Set Summary

POINTER ADDRESS			POWER-ON RESET		
HEX	REGISTER NAME	FUNCTION	BINARY	HEX	TYPE(1)
00h	Configuration Register	All-register reset, shunt voltage and bus voltage ADC conversion times and averaging, operating mode.	01000001 00100111	4127	R/ W
01h	Shunt Voltage Register	Shunt voltage measurement data.	00000000 00000000	0000	R
02h	Bus Voltage Register	Bus voltage measurement data.	00000000 00000000	0000	R
03h	Power Register(2)	Contains the value of the calculated power being delivered to the load.	00000000 00000000	0000	R
04h	Current Register(2)	Contains the value of the calculated current flowing through the shunt resistor.	00000000 00000000	0000	R
05h	Calibration Register	Sets full-scale range and LSB of current and power measurements. Overall system calibration.	00000000 00000000	0000	R/ W
06h	Mask/Enable Register	Alert configuration and Conversion Ready flag.	00000000 00000000	0000	R/ W
07h	Alert Limit Register	Contains the limit value to compare to the selected Alert function.	00000000 00000000	0000	R/ W
FEh	Manufacturer ID Register	Contains unique manufacturer identification number.	0101010001001001	5449	R
FFh	Die ID Register	Contains unique die identification number.	0010001001100000	2260	R

(1) Type: **R** = Read-Only, **R/ W** = Read/Write.
(2) The Current Register (04h) and Power Register (03h) default to '0' because the Calibration register defaults to '0', yielding zero current and power values until the Calibration register is programmed.

7.1.1 Configuration Register (00h) (Read/Write)

Table 7-2. Configuration Register (00h) (Read/Write) Descriptions

BIT NO.	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	RST	—	—	—	AVG2	AVG1	AVG0	VBUSCT2	VBUSCT1	VBUSCT0	VSHCT2	VSHCT1	VSHCT0	MODE3	MODE2	MODE1
POR VALUE	0	1	0	0	0	0	0	1	0	0	1	0	0	1	1	1

The Configuration Register settings control the operating modes for the device. This register controls the conversion time settings for both the shunt and bus voltage measurements as well as the averaging mode used. The operating mode that controls what signals are selected to be measured is also programmed in the Configuration Register .

The Configuration Register can be read from at any time without impacting or affecting the device settings or a conversion in progress. Writing to the Configuration Register halts any conversion in progress until the write sequence is completed resulting in a new conversion starting based on the new contents of the Configuration Register (00h). This halt prevents any uncertainty in the conditions used for the next completed conversion.

RST: **Reset Bit**

Bit 15 Setting this bit to '1' generates a system reset that is the same as power-on reset. Resets all registers to default values; this bit self-clears.

AVG: Averaging Mode

Bits 9–11 Determines the number of samples that are collected and averaged. Table 7-3 shows all the AVG bit settings and related number of averages for each bit setting.

Table 7-3. AVG Bit Settings[11:9] Combinations

AVG2 D11	AVG1 D10	AVG0 D9	NUMBER OF AVERAGES(1)
0	0	0	1
0	0	1	4
0	1	0	16
0	1	1	64
1	0	0	128
1	0	1	256
1	1	0	512
1	1	1	1024

(1) Shaded values are default.

VBUSCT: Bus Voltage Conversion Time

Bits 6–8 Sets the conversion time for the bus voltage measurement. Table 7-4 shows the VBUSCT bit options and related conversion times for each bit setting.

Table 7-4. VBUSCT Bit Settings [8:6] Combinations

VBUSCT2 D8	VBUSCT1 D7	VBUSCT0 D6	CONVERSION TIME(1)
0	0	0	140 µs
0	0	1	204 µs
0	1	0	332 µs
0	1	1	588 µs
1	0	0	1.1 ms
1	0	1	2.116 ms
1	1	0	4.156 ms
1	1	1	8.244 ms

(1) Shaded values are default.

VSHCT: Shunt Voltage Conversion Time

Bits 3–5 Sets the conversion time for the shunt voltage measurement. Table 7-5 shows the VSHCT bit options and related conversion times for each bit setting.

Table 7-5. VSHCT Bit Settings [5:3] Combinations

VSHCT2 D8	VSHCT1 D7	VSHCT0 D6	CONVERSION TIME(1)
0	0	0	140 µs
0	0	1	204 µs
0	1	0	332 µs
0	1	1	588 µs
1	0	0	1.1 ms
1	0	1	2.116 ms
1	1	0	4.156 ms
1	1	1	8.244 ms

(1) Shaded values are default.

MODE: Operating Mode

Bits 0-2 — Selects continuous, triggered, or power-down mode of operation. These bits default to continuous shunt and bus measurement mode. The mode settings are shown in Table 7-6.

Table 7-6. Mode Settings [2:0] Combinations

MODE3 D2	MODE2 D1	MODE1 D0	MODE(1)
0	0	0	Power-Down (or Shutdown)
0	0	1	Shunt Voltage, Triggered
0	1	0	Bus Voltage, Triggered
0	1	1	Shunt and Bus, Triggered
1	0	0	Power-Down (or Shutdown)
1	0	1	Shunt Voltage, Continuous
1	1	0	Bus Voltage, Continuous
1	1	1	Shunt and Bus, Continuous

(1) Shaded values are default.

7.1.2 Shunt Voltage Register (01h) (Read-Only)

The Shunt Voltage Register stores the current shunt voltage reading, V_{SHUNT}. Negative numbers are represented in two's complement format. Generate the two's complement of a negative number by complementing the absolute value binary number and adding 1. An MSB = '1' denotes a negative number.

Example: For a value of V_{SHUNT} = –80 mV:

1. Take the absolute value: 80 mV
2. Translate this number to a whole decimal number (80 mV ÷ 2.5 µV) = 32000
3. Convert this number to binary = 0111 1101 0000 0000
4. Complement the binary result = 1000 0010 1111 1111
5. Add '1' to the complement to create the two's complement result = 1000 0011 0000 0000 = 8300h

If averaging is enabled, this register displays the averaged value.

Full-scale range = 81.92 mV (decimal = 7FFF); LSB: 2.5 µV.

Table 7-7. Shunt Voltage Register (01h) (Read-Only) Description

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	SIGN	SD14	SD13	SD12	SD11	SD10	SD9	SD8	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.1.3 Bus Voltage Register (02h) (Read-Only) (1)

The Bus Voltage Register stores the most recent bus voltage reading, VBUS.

If averaging is enabled, this register displays the averaged value.

Full-scale range = 40.96 V (decimal = 7FFF); LSB = 1.25 mV.

Table 7-8. Bus Voltage Register (02h) (Read-Only) Description

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	—	BD14	BD13	BD12	BD11	BD10	BD9	BD8	BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(1) D15 is always zero because bus voltage can only be positive.

7.1.4 Power Register (03h) (Read-Only)

If averaging is enabled, this register displays the averaged value.

The Power Register LSB is internally programmed to equal 25 times the programmed value of the Current_LSB.

The Power Register records power in Watts by multiplying the decimal values of the Current Register with the decimal value of the Bus Voltage Register according to Equation 4.

Table 7-9. Power Register (03h) (Read-Only) Description

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.1.5 Current Register (04h) (Read-Only)

If averaging is enabled, this register displays the averaged value.

The value of the Current Register is calculated by multiplying the decimal value in the Shunt Voltage Register with the decimal value of the Calibration Register, according to Equation 3.

Table 7-10. Current Register (04h) (Read-Only) Register Description

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	CSIGN	CD14	CD13	CD12	CD11	CD10	CD9	CD8	CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.1.6 Calibration Register (05h) (Read/Write)

This register provides the device with the value of the shunt resistor that is present to create the measured differential voltage. The register also sets the resolution of the Current Register. Programming this register sets the Current_LSB and the Power_LSB. This register is also used in overall system calibration. See the *Programming the Calibration Register* for additional information on programming the Calibration Register.

Table 7-11. Calibration Register (05h) (Read/Write) Description

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	—	FS14	FS13	FS12	FS11	FS10	FS9	FS8	FS7	FS6	FS5	FS4	FS3	FS2	FS1	FS0
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.1.7 Mask/Enable Register (06h) (Read/Write)

The Mask/Enable Register selects the function that is enabled to control the Alert pin as well as how that pin functions. If multiple functions are enabled, the highest significant bit position Alert Function (D15-D11) takes priority and responds to the Alert Limit Register.

Table 7-12. Mask/Enable Register (06h) (Read/Write)

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	SOL	SUL	BOL	BUL	POL	CNVR	—	—	—	—	—	AFF	CVRF	OVF	APOL	LEN
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SOL: **Shunt Voltage Over-Voltage**

Bit 15 Setting this bit high configures the Alert pin to be asserted if the shunt voltage measurement following a conversion exceeds the value programmed in the Alert Limit Register.

SUL: **Shunt Voltage Under-Voltage**

Bit 14	Setting this bit high configures the Alert pin to be asserted if the shunt voltage measurement following a conversion drops below the value programmed in the Alert Limit Register.
BOL:	**Bus Voltage Over-Voltage**
Bit 13	Setting this bit high configures the Alert pin to be asserted if the bus voltage measurement following a conversion exceeds the value programmed in the Alert Limit Register.
BUL:	**Bus Voltage Under-Voltage**
Bit 12	Setting this bit high configures the Alert pin to be asserted if the bus voltage measurement following a conversion drops below the value programmed in the Alert Limit Register.
POL:	**Power Over-Limit**
Bit 11	Setting this bit high configures the Alert pin to be asserted if the Power calculation made following a bus voltage measurement exceeds the value programmed in the Alert Limit Register.
CNVR:	**Conversion Ready**
Bit 10	Setting this bit high configures the Alert pin to be asserted when the Conversion Ready Flag, Bit 3, is asserted indicating that the device is ready for the next conversion.
AFF:	**Alert Function Flag**
Bit 4	While only one Alert Function can be monitored at the Alert pin at a time, the Conversion Ready can also be enabled to assert the Alert pin. Reading the Alert Function Flag following an alert allows the user to determine if the Alert Function is the source of the Alert. When the Alert Latch Enable bit is set to Latch mode, the Alert Function Flag bit clears only when the Mask/Enable Register is read. When the Alert Latch Enable bit is set to Transparent mode, the Alert Function Flag bit is cleared following the next conversion that does not result in an Alert condition.
CVRF:	**Conversion Ready Flag**
Bit 3	Although the device can be read at any time, and the data from the last conversion is available, the Conversion Ready Flag bit is provided to help coordinate one-shot or triggered conversions. The Conversion Ready Flag bit is set after all conversions, averaging, and multiplications are complete. Conversion Ready Flag bit clears under the following conditions: 1.) Writing to the Configuration Register (except for Power-Down selection) 2.) Reading the Mask/Enable Register
OVF:	**Math Overflow Flag**
Bit 2	This bit is set to '1' if an arithmetic operation resulted in an overflow error. The bit indicates that current and power data can be invalid.
APOL:	**Alert Polarity bit; sets the Alert pin polarity.**
Bit 1	1 = Inverted (active-high open collector) 0 = Normal (active-low open collector) (default)
LEN:	**Alert Latch Enable; configures the latching feature of the Alert pin and Alert Flag bits.**
Bit 0	1 = Latch enabled 0 = Transparent (default) When the Alert Latch Enable bit is set to Transparent mode, the Alert pin and Flag bit resets to the idle states when the fault has been cleared. When the Alert Latch Enable bit is set to Latch mode, the Alert pin and Alert Flag bit remains active following a fault until the Mask/Enable Register has been read.

7.1.8 Alert Limit Register (07h) (Read/Write)

The Alert Limit Register contains the value used to compare to the register selected in the Mask/Enable Register to determine if a limit has been exceeded.

Table 7-13. Alert Limit Register (07h) (Read/Write) Description

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	AUL15	AUL14	AUL13	AUL12	AUL11	AUL10	AUL9	AUL8	AUL7	AUL6	AUL5	AUL4	AUL3	AUL2	AUL1	AUL0
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.1.9 Manufacturer ID Register (FEh) (Read-Only)

The Manufacturer ID Register stores a unique identification number for the manufacturer.

Table 7-14. Manufacturer ID Register (FEh) (Read-Only) Description

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
POR VALUE	0	1	0	1	0	1	0	0	0	1	0	0	1	0	0	1

ID:	Manufacturer ID Bits
Bits 0-15	Stores the manufacturer identification bits

7.1.10 Die ID Register (FFh) (Read-Only)

The Die ID Register stores a unique identification number and the revision ID for the die.

Table 7-15. Die ID Register (FFh) (Read-Only) Description

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	DID11	DID10	DID9	DID8	DID7	DID6	DID5	DID4	DID3	DID2	DID1	DID0	RID3	RID2	RID1	RID0
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DID:	Device ID Bits
Bits 4-15	Stores the device identification bits

RID:	Die Revision ID Bits
Bit 0-3	Stores the device revision identification bits

8 Application and Implementation

Note

Information in the following applications sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes, as well as validating and testing their design implementation to confirm system functionality.

8.1 Application Information

The INA226 is a current shunt and power monitor with an I2C™ compatible interface. The device monitors both a shunt voltage drop and bus supply voltage. Programmable calibration value, conversion times, and averaging, combined with an internal multiplier, enable direct readouts of current in amperes and power in watts.

8.2 Typical Applications

8.2.1 High-Side Sensing Circuit Application

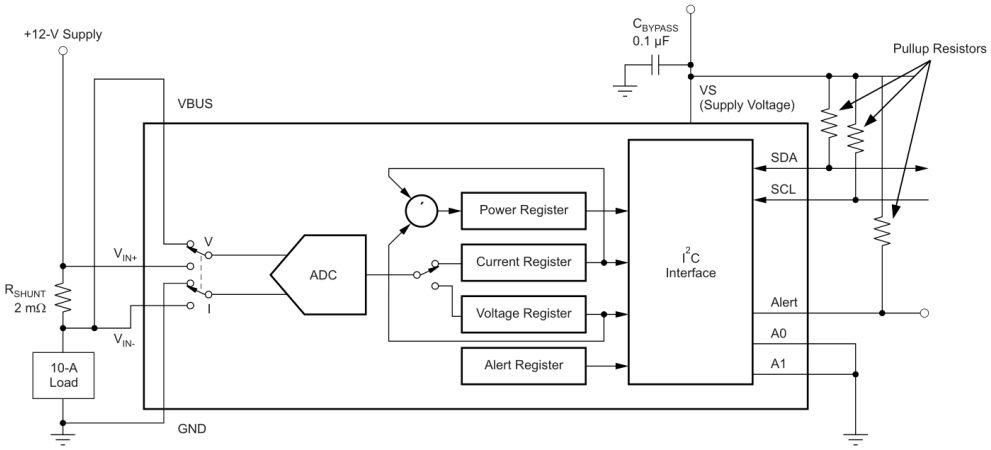


Figure 8-1. Typical Circuit Configuration, INA226

8.2.1.1 Design Requirements

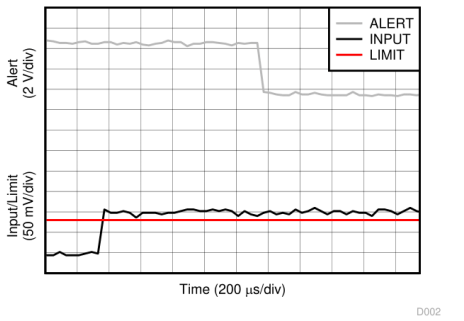
INA226 measures the voltage developed across a current-sensing resistor (R_{SHUNT}) when current passes through the resistor. The device also measures the bus supply voltage and can calculate power when calibrated. The device comes with alert capability where the alert pin can be programmed to respond to a user-defined event or to a conversion ready notification. This design illustrates the ability of the alert pin to respond to a set threshold.

8.2.1.2 Detailed Design Procedure

The Alert pin can be configured to respond to one of the five alert functions described in the *Alert Pin* section. The alert pin must to be pulled up to the V_{VS} pin voltage via the pull-up resistors. The configuration register is set based on the required conversion time and averaging. The Mask/Enable Register is set to identify the required alert function and the Alert Limit Register is set to the limit value used for comparison.

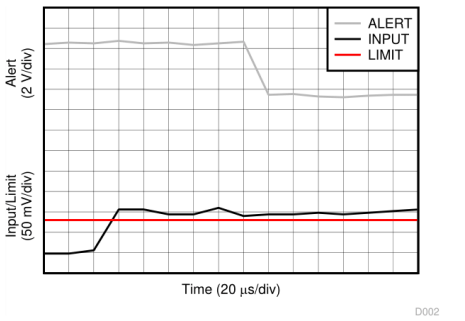
8.2.1.3 Application Curves

Figure 8-2 shows the Alert pin response to a shunt voltage over-limit of 80 mV for a conversion time (t_{CT}) of 1.1 ms and averaging set to 1. Figure 8-3 shows the response for the same limit but with the conversion time reduced to 140 µs.



See Table 8-1 t_{CT} = 1.1 ms
See Table 8-3 See Table 8-4

Figure 8-2. Alert Response



See Table 8-2 t_{CT} = 140 µs
See Table 8-3 See Table 8-4

Figure 8-3. Alert Response

Table 8-1. Configuration Register (00h) Settings for Figure 8-2 (Value = 4025h)

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	RST	—	—	—	AVG2	AVG1	AVG0	VBUSCT 2	VBUSCT 1	VBUSCT 0	V_{SH}CT2	V_{SH}CT1	V_{SH}CT0	MODE3	MODE2	MODE1
POR VALUE	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1

Table 8-2. Configuration Register (00h) Settings for Figure 8-3 (Value = 4005h)

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	RST	—	—	—	AVG2	AVG1	AVG0	VBUSCT 2	VBUSCT 1	VBUSCT 0	V_{SH}CT2	V_{SH}CT1	V_{SH}CT0	MODE3	MODE2	MODE1
POR VALUE	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Table 8-3. Mask/Enable Register (06h) Settings for Figure 8-2 and Figure 8-3 (Value = 8000h)

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	SOL	SUL	BOL	BUL	POL	CNVR	—	—	—	—	—	AFF	CVRF	OVF	APOL	LEN
POR VALUE	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8-4. Alert Limit Register (07h) Settings for Figure 8-2 and Figure 8-3 (Value = 7D00)

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	AUL15	AUL14	AUL13	AUL12	AUL11	AUL10	AUL9	AUL8	AUL7	AUL6	AUL5	AUL4	AUL3	AUL2	AUL1	AUL0
POR VALUE	0	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0

8.3 Power Supply Recommendations

The input circuitry of the device can accurately measure signals on common-mode voltages beyond the power supply voltage, V_{VS}. For example, the voltage applied to the V_{VS} power supply terminal can be 5 V, whereas the load power-supply voltage being monitored (the common-mode voltage) can be as high as 36 V. Note also that the device can withstand the full 0-V to 36-V range at the input terminals, regardless of whether the device has power applied or not.

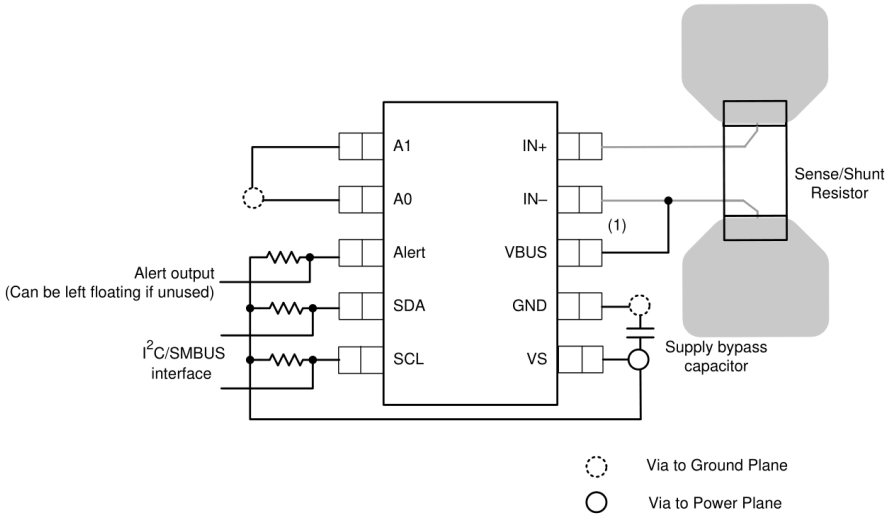
Place the required power-supply bypass capacitors as close as possible to the supply and ground terminals of the device to provide stability. A typical value for this supply bypass capacitor is 0.1 µF. Applications with noisy or high-impedance power supplies can require additional decoupling capacitors to reject power-supply noise.

8.4 Layout

8.4.1 Layout Guidelines

Connect the input pins (IN+ and IN-) to the sensing resistor using a Kelvin connection or a 4-wire connection. These connection techniques verify that only the current-sensing resistor impedance is detected between the input pins. Poor routing of the current-sensing resistor commonly results in additional resistance present between the input pins. Given the very low ohmic value of the current-sensing resistor, any additional high-current carrying impedance causes significant measurement errors. Place the power-supply bypass capacitor as close as possible to the supply and ground pins.

8.4.2 Layout Example



(1) connect the VBUS pin to the power supply rail.

Figure 8-4. INA226 Layout Example

9 Device and Documentation Support

9.1 Device Support

9.1.1 Development Support

- INA226EVM Evaluation Board and Software Tutorial (SBOU113)

9.2 Receiving Notification of Documentation Updates

To receive notification of documentation updates, navigate to the device product folder on ti.com. Click on *Notifications* to register and receive a weekly digest of any product information that has changed. For change details, review the revision history included in any revised document.

9.3 Support Resources

TI E2E™ support forums are an engineer's go-to source for fast, verified answers and design help — straight from the experts. Search existing answers or ask your own question to get the quick design help you need.

Linked content is provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

9.4 Trademarks

I2C™ is a trademark of I2C Technologies, Ltd.
TI E2E™ is a trademark of Texas Instruments.
All trademarks are the property of their respective owners.

9.5 Electrostatic Discharge Caution

This integrated circuit can be damaged by ESD. Texas Instruments recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage.

ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications.

9.6 Glossary

TI Glossary	This glossary lists and explains terms, acronyms, and definitions.

10 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision A (August 2015) to Revision B (September 2024)	Page
• Updated the number formatting for tables, figures, and cross-references throughout the document	1
• Removed Machine Model from ESD Ratings	4
• Updated thermal metric values to match thermal model	4
• Added ± in front of typical values for PSRR specifications	5
• Decreased input bias current typical value	5
• Updated *Shunt Input Gain Error vs Common-Mode Voltage* graph and Input Bias current curves	7

Changes from Revision * (June 2011) to Revision A (August 2015)	Page
• Added Handling Rating table, Feature Description section, Device Functional Modes, Application and Implementation section, Power Supply Recommendations section, Layout section, Device and Documentation Support section, and Mechanical, Packaging, and Orderable Information section	1

11 Mechanical, Packaging, and Orderable Information

The following pages include mechanical, packaging, and orderable information. This information is the most current data available for the designated devices. This data is subject to change without notice and revision of this document. For browser-based versions of this data sheet, refer to the left-hand navigation.

PACKAGING INFORMATION

Orderable part number	Status (1)	Material type (2)	Package | Pins	Package qty | Carrier	RoHS (3)	Lead finish/ Ball material (4)	MSL rating/ Peak reflow (5)	Op temp (°C)	Part marking (6)
INA226AIDGSR	Active	Production	VSSOP (DGS) | 10	2500 | LARGE T&R	Yes	NIPDAU | SN | NIPDAUAG	Level-2-260C-1 YEAR	-40 to 125	226
INA226AIDGST	Active	Production	VSSOP (DGS) | 10	250 | SMALL T&R	Yes	NIPDAU | SN | NIPDAUAG	Level-2-260C-1 YEAR	-40 to 125	226

(1) **Status:** For more details on status, see our product life cycle.

(2) **Material type:** When designated, preproduction parts are prototypes/experimental devices, and are not yet approved or released for full production. Testing and final process, including without limitation quality assurance, reliability performance testing, and/or process qualification, may not yet be complete, and this item is subject to further changes or possible discontinuation. If available for ordering, purchases will be subject to an additional waiver at checkout, and are intended for early internal evaluation purposes only. These items are sold without warranties of any kind.

(3) **RoHS values:** Yes, No, RoHS Exempt. See the TI RoHS Statement for additional information and value definition.

(4) **Lead finish/Ball material:** Parts may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead finish/Ball material values may wrap to two lines if the finish value exceeds the maximum column width.

(5) **MSL rating/Peak reflow:** The moisture sensitivity level ratings and peak solder (reflow) temperatures. In the event that a part has multiple moisture sensitivity ratings, only the lowest level per JEDEC standards is shown. Refer to the shipping label for the actual reflow temperature that will be used to mount the part to the printed circuit board.

(6) **Part marking:** There may be an additional marking, which relates to the logo, the lot trace code information, or the environmental category of the part.

Multiple part markings will be inside parentheses. Only one part marking contained in parentheses and separated by a "~" will appear on a part. If a line is indented then it is a continuation of the previous line and the two combined represent the entire part marking for that device.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

OTHER QUALIFIED VERSIONS OF INA226 :

PACKAGE OPTION ADDENDUM

www.ti.com 7-May-2025

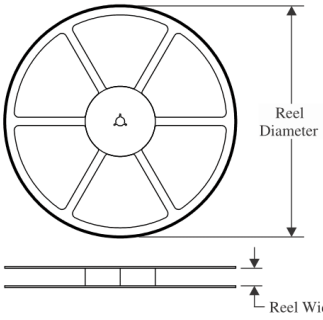
- Automotive : INA226-Q1

NOTE: Qualified Version Definitions:

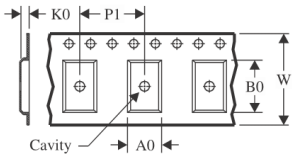
- Automotive - Q100 devices qualified for high-reliability automotive applications targeting zero defects

TAPE AND REEL INFORMATION

REEL DIMENSIONS

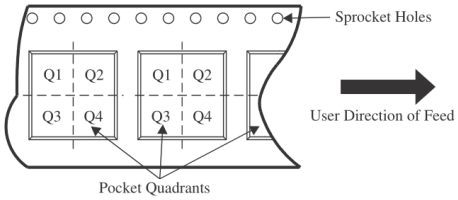


TAPE DIMENSIONS



A0	Dimension designed to accommodate the component width
B0	Dimension designed to accommodate the component length
K0	Dimension designed to accommodate the component thickness
W	Overall width of the carrier tape
P1	Pitch between successive cavity centers

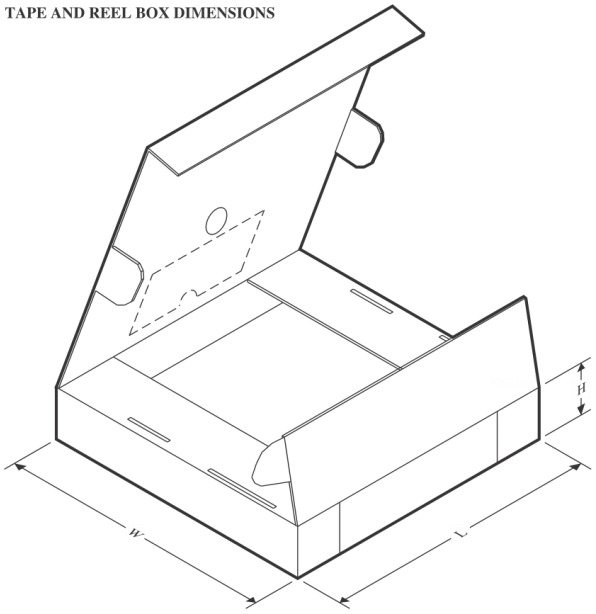
QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE



*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
INA226AIDGSR	VSSOP	DGS	10	2500	330.0	12.4	5.3	3.4	1.4	8.0	12.0	Q1
INA226AIDGST	VSSOP	DGS	10	250	330.0	12.4	5.25	3.35	1.25	8.0	12.0	Q1

TAPE AND REEL BOX DIMENSIONS



*All dimensions are nominal

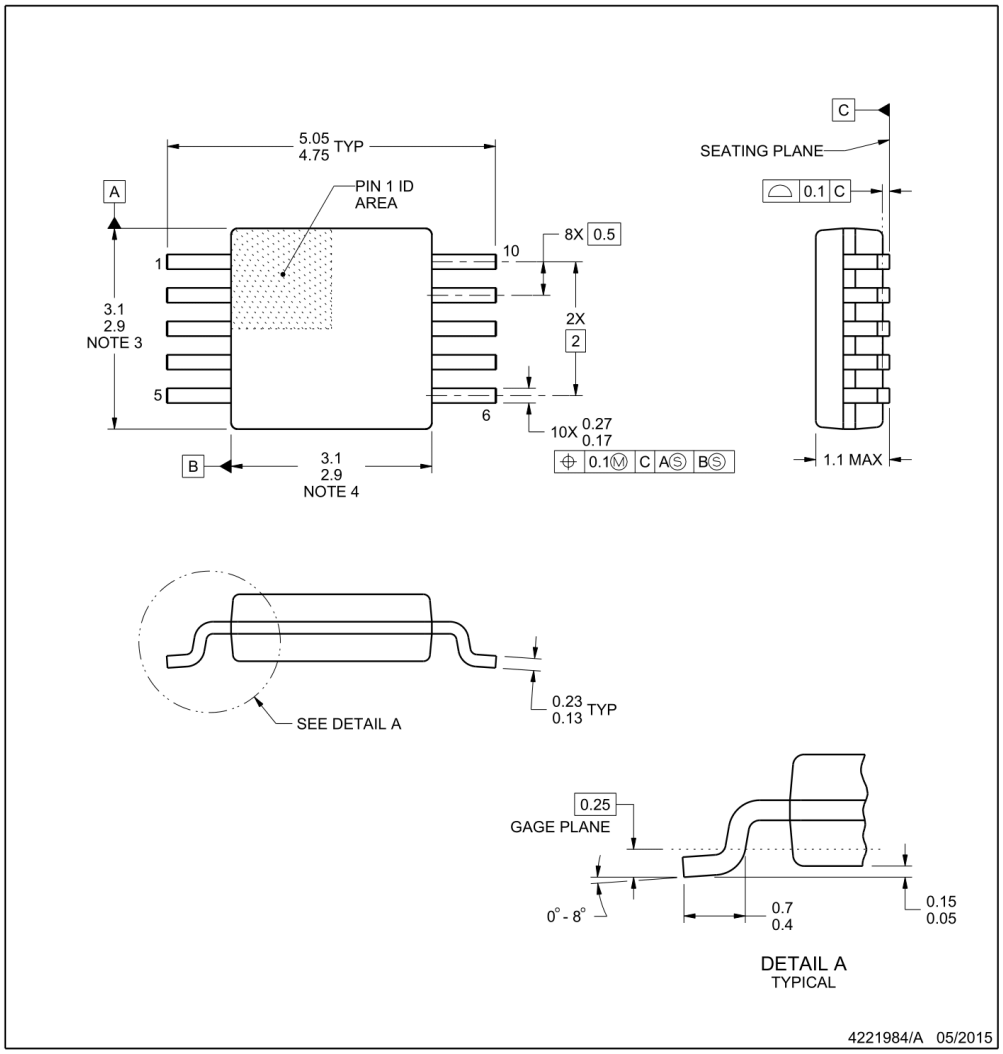
Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
INA226AIDGSR	VSSOP	DGS	10	2500	353.0	353.0	32.0
INA226AIDGST	VSSOP	DGS	10	250	366.0	364.0	50.0

PACKAGE OUTLINE

DGS0010A

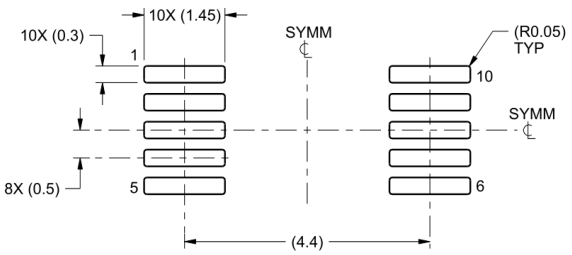
VSSOP - 1.1 mm max height

SMALL OUTLINE PACKAGE

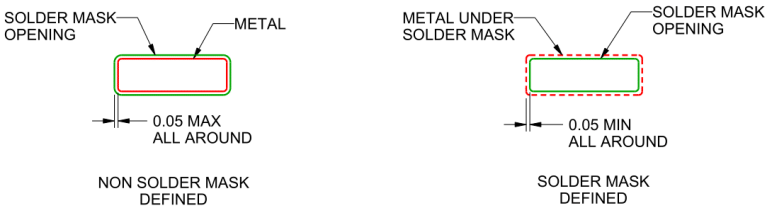


NOTES:

1. All linear dimensions are in millimeters. Any dimensions in parenthesis are for reference only. Dimensioning and tolerancing per ASME Y14.5M.
2. This drawing is subject to change without notice.
3. This dimension does not include mold flash, protrusions, or gate burrs. Mold flash, protrusions, or gate burrs shall not exceed 0.15 mm per side.
4. This dimension does not include interlead flash. Interlead flash shall not exceed 0.25 mm per side.
5. Reference JEDEC registration MO-187, variation BA.



LAND PATTERN EXAMPLE
SCALE:10X

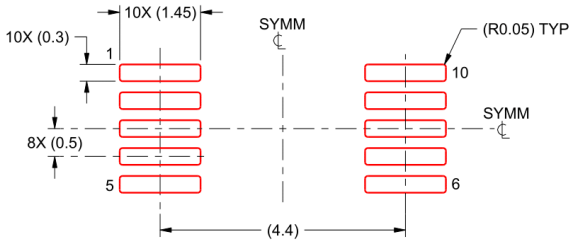


SOLDER MASK DETAILS
NOT TO SCALE

4221984/A 05/2015

NOTES: (continued)

6. Publication IPC-7351 may have alternate designs.
7. Solder mask tolerances between and around signal pads can vary based on board fabrication site.



SOLDER PASTE EXAMPLE
BASED ON 0.125 mm THICK STENCIL
SCALE:10X

4221984/A 05/2015

NOTES: (continued)

8. Laser cutting apertures with trapezoidal walls and rounded corners may offer better paste release. IPC-7525 may have alternate design recommendations.
9. Board assembly site may have different recommendations for stencil design.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Appendix H – Jetson Nano Datasheet

DATA SHEET

NVIDIA Jetson Nano System-on-Module

Maxwell GPU + ARM Cortex-A57 + 4GB LPDDR4 + 16GB eMMC

Maxwell GPU ◊

128-core GPU | End-to-end lossless compression | Tile Caching | OpenGL® 4.6 | OpenGL ES 3.2 | Vulkan™ 1.1 | CUDA® | OpenGL ES Shader Performance (up to): 512 GFLOPS (FP16)
Maximum Operating Frequency: 921MHz

CPU

ARM® Cortex® -A57 MPCore (Quad-Core) Processor with NEON Technology | L1 Cache: 48KB L1 instruction cache (I-cache) per core; 32KB L1 data cache (D-cache) per core | L2 Unified Cache: 2MB | Maximum Operating Frequency: 1.43GHz

Audio

Industry standard High Definition Audio (HDA) controller provides a multichannel audio path to the HDMI interface.

Memory

Dual Channel | System MMU | Memory Type: 4ch x 16-bit LPDDR4 | Maximum Memory Bus Frequency: 1600MHz | Peak Bandwidth: 25.6 GB/s | Memory Capacity: 4GB

Storage

eMMC 5.1 Flash Storage | Bus Width: 8-bit | Maximum Bus Frequency: 200MHz (HS400) | Storage Capacity: 16GB

Boot Sources

eMMC and USB (recovery mode)

Networking

10/100/1000 BASE-T Ethernet | Media Access Controller (MAC)

Imaging

Dedicated RAW to YUV processing engines process up to 1400Mpix/s (up to 24MP sensor) | MIPI CSI 2.0 up to 1.5Gbps (per lane) | Support for x4 and x2 configurations (up to four active streams).

Operating Requirements

Temperature Range (T_j): -25 – 97C* | Module Power: 5 – 10W | Power Input: 5.0V

Display Controller

Two independent display controllers support DSI, HDMI, DP, eDP:

- MIPI-DSI (1.5Gbps/lane): Single x2 lane | Maximum Resolution: 1920x960 at 60Hz (up to 24bpp)
- HDMI 2.0a/b (up to 6Gbps) | DP 1.2a (HBR2 5.4 Gbps) | eDP 1.4 (HBR2 5.4Gbps) | Maximum Resolution (DP/eDP/HDMI): 3840 x 2160 at 60Hz (up to 24bpp)

Clocks

System clock: 38.4MHz | Sleep clock: 32.768kHz | Dynamic clock scaling and clock source selection

Multi-Stream HD Video and JPEG

Video Decode

- H.265 (Main, Main 10): 2160p 60fps | 1080p 240fps
- H.264 (BP/MP/HP/Stereo SEI half-res): 2160p 60fps | 1080p 240fps
- H.264 (MVC Stereo per view): 2160p 30fps | 1080p 120fps
- VP9 (Profile 0, 8-bit): 2160p 60fps | 1080p 240fps
- VP8: 2160p 60fps | 1080p 240fps
- VC-1 (Simple, Main, Advanced): 1080p 120fps | 1080i 240fps
- MPEG-2 (Main): 2160p 60fps | 1080p 240fps | 1080i 240fps

Video Encode

- H.265:2160p 30fps | 1080p 120fps
- H.264 (BP/MP/HP): 2160p 30fps | 1080p 120fps
- H.264 (MVC Stereo per view): 1440p 30fps | 1080p 60fps
- VP8: 2160p 30fps | 1080p 120fps

JPEG (Decode and Encode): 600 MP/s

Peripheral Interfaces

xHCI host controller with integrated PHY: 1 x USB 3.0, 3 x USB 2.0 | USB 3.0 device controller with integrated PHY | EHCI controller with embedded hub for USB 2.0 | 4-lane PCIe: one x1/2/4 controller | single SD/MMC controller (supporting SDIO 4.0, SD HOST 4.0) | 3 x UART | 2 x SPI | 4 x I2C | 2 x I2S: support I2S, RJM, LJM, PCM, TDM (multi-slot mode) | GPIOs

Mechanical

Module Size: 69.6 mm x 45 mm | PCB: 8L HDI | Connector: 260 pin SO-DIMM

Note: Refer to the software release feature list for current software support; all features may not be available for a particular OS.

◊ Product is based on a published Khronos Specification and is expected to pass the Khronos Conformance Process. Current conformance status can be found at www.khronos.org/conformance.

* See the *Jetson Nano Thermal Design Guide* for details. Listed temperature range is based on module T_j characterization.

Revision History

Version	Date	Description
v0.1	JAN 2019	Initial Release
v0.7	MAY 2019	Description • Memory: corrected peak bandwidth • Peripheral Interfaces: corrected number of available I2C interfaces Functional Overview • Removed block diagram; see the *Jetson Nano Product Design Guide* for these details Power and System Management • Removed On-Module Internal Power Rails table • Updated Power Domains table • Updated Programmable Interface Wake Event table • Updated Power Up/Down sequence diagrams Pin Descriptions • Updated throughout to reflect updated pinmux • GPIO Pins: updated table to reflect dedicated GPIO pins only (see pinmux for ALL GPIO capable pins) Interface Descriptions • Updated throughout to reflect updated pinmux • Embedded DisplayPort (eDP) Interface: clarified DP use/limitations on DP0 • MIPI Camera Serial Interface (CSI) - Updated CSI description to remove erroneous reference to virtual channels Physical/Electrical Characteristics • Absolute Maximum Ratings - Added reference to Jetson Nano Thermal Design Guide for Operating Temperature; extended IDD_{MAX} to 5A • Pinout: Updated to reflect updated pinmux • Package Drawing and Dimensions – Updated drawing
v0.8	OCT 2019	Description • Operating Requirements: corrected Module Power to reflect power for module only (previous stated range included module + IO); updated Temperature Range for clarity, included maximum operating temperature and updated note to reflect module temperature is based on T_j.
v1.0	FEB 2020	Pin Descriptions • GPIO Pins: corrected pin number listing for GPIO01 Interface Descriptions • High-Definition Multimedia Interface (HDMI) and DisplayPort (DP) Interfaces reference to YUV output support • Gigabit Ethernet – Corrected Realtek Gigabit Ethernet Controller part number Physical/Electrical Characteristics • Operating and Absolute Maximum Ratings – Added Mounting Force to Absolute Maximum Ratings table. • Package Drawing and Dimensions – Updated drawing • Environmental & Mechanical Screening – Added section
v1.1	MAY 2022	Power and System Management • Moved PMIC_BBAT information to new sub-section • Updated PMIC_BBAT Pin Description Pin Descriptions • GPIO Pins: updated GPIO8 (pin 208) description; Fan tachometer only Interface Descriptions • SD/SDIO – Updated pin descriptions to include 3.3V support; deprecated GPIO8 SD Card Detect support • SPI – updated master timing diagram and parameters • UART – UART1_CTS (pin 209) updated PoR Physical/Electrical Characteristics • Package Drawing and Dimensions – added module dimensions table

Table of Contents

1.0 Functional Overview

Designed for use in power-limited environments, the Jetson Nano squeezes industry-leading compute capabilities, 64-bit operating capability, and integrated advanced multi-function audio, video and image processing pipelines into a 260-pin SO-DIMM. The Maxwell GPU architecture implemented several architectural enhancements designed to extract maximum performance per watt consumed. Core components of the Jetson Nano series module include:

- NVIDIA® Tegra® X1 series SoC
  - NVIDIA Maxwell GPU
  - ARM® quad-core Cortex®-A57 CPU Complex
- 4GB LPDDR4 memory
- 16GB eMMC 5.1 storage
- Gigabit Ethernet (10/100/1000 Mbps)
- PMIC, regulators, power and voltage monitors
- 260-pin keyed connector (exposes both high-speed and low-speed industry standard I/O)
- On-chip temperature sensors

1.1 Maxwell GPU

The Graphics Processing Cluster (GPC) is a dedicated hardware block for rasterization, shading, texturing, and compute; most of the GPU's core graphics functions are performed inside the GPC. Within the GPC there are multiple Streaming Multiprocessor (SM) units and a Raster Engine. Each SM includes a Polymorph Engine and Texture Units; raster operations remain aligned with L2 cache slices and memory controllers

The Maxwell GPU architecture introduced an all-new design for the SM, redesigned all unit and crossbar structures, optimized data flows, and significantly improved power management. The SM scheduler architecture and algorithms were rewritten to be more intelligent and avoid unnecessary stalls, while further reducing the energy per instruction required for scheduling. The organization of the SM also changed; each Maxwell SM (called SMM) is now partitioned into four separate processing blocks, each with its own instruction buffer, scheduler and 32 CUDA cores.

The SMM CUDA cores perform pixel/vertex/geometry shading and physics/compute calculations. Texture units perform texture filtering and load/store units fetch and save data to memory. Special Function Units (SFUs) handle transcendental and graphics interpolation instructions. Finally, the Polymorph Engine handles vertex fetch, tessellation, viewport transform, attribute setup, and stream output. The SMM geometry and pixel processing performance make it highly suitable for rendering advanced user interfaces and complex gaming applications; the power efficiency of the Maxwell GPU enables this performance on devices with power-limited environments.

Features:

- End-to-end lossless compression
- Tile Caching
- Support for OpenGL 4.6, OpenGL ES 3.2, Vulkan 1.1, DirectX 12, CUDA 10 (FP16)
- Adaptive Scalable Texture Compression (ATSC) LDR profile supported
- Iterated blend, ROP OpenGL-ES blend modes
- 2D BLIT from 3D class avoids channel switch
- 2D color compression
- Constant color render SM bypass
- 2x, 4x, 8x MSAA with color and Z compression
- Non-power-of-2 and 3D textures, FP16 texture filtering

- FP16 shader support
- Geometry and Vertex attribute Instancing
- Parallel pixel processing
- Early-z reject: Fast rejection of occluded pixels acts as multiplier on pixel shader and texture performance while saving power and bandwidth
- Video protection region
- Power saving: Multiple levels of clock gating for linear scaling of power

GPU frequency and voltage are actively managed by Tegra Power and Thermal Management Software and influenced by workload. Frequency may be throttled at higher temperatures (above a specified threshold) resulting in a behavior that reduces the GPU operating frequency. Observed chip-to-chip variance is due to NVIDIA ability to maximize performance (DVFS) on a per-chip basis, within the available power budget.

1.2 CPU Complex

The CPU complex is a high-performance Multi-Core SMP cluster of four ARM Cortex-A57 CPUs with 2MB of L2 cache (shared by all cores). Features include:

- Superscalar, variable-length, out-of-order pipeline
- Dynamic branch prediction with Branch Target Buffer (BTB) and Global History Buffer RAMs, a return stack, and an indirect predictor
- 48-entry fully-associative L1 instruction TLB with native support for 4KB, 64KB, and 1MB page sizes.
- 32-entry fully-associative L1 data TLB with native support for 4KB, 64KB, and 1MB pages sizes.
- 4-way set-associative unified 1024-entry Level 2 (L2) TLB in each processor
- 48Kbyte I-cache and 32Kbyte D-cache for each core.
- Full implementation of ARMv8 architecture instruction set
- Embedded Trace Microcell (ETM) based on the ETMv4 architecture
- Performance Monitor Unit (PMU) based on the PMUv3 architecture
- Cross Trigger Interface (CTI) for multiprocessor debugging
- Cryptographic Engine for crypto function support
- Interface to an external Generic Interrupt Controller (vGIC-400)
- Power management with multiple power domains

CPU frequency and voltage are actively managed by Tegra Power and Thermal Management Software and influenced by workload. Frequency may be throttled at higher temperatures (above a specified threshold) resulting in a behavior that reduces the CPU operating frequency. Observed chip-to-chip variance is due to NVIDIA ability to maximize performance (DVFS) on a per-chip basis, within the available power budget.

1.2.1 Snoop Control Unit and L2 Cache

The CPU cluster includes an integrated snoop control unit (SCU) that maintains coherency between the CPUs within the cluster and a tightly coupled L2 cache that is shared between the CPUs within the cluster. The L2 cache also provides a 128-bit AXI master interface to access DRAM. L2 cache features include:

- 2MB L2
- Fixed line length of 64 bytes
- 16-way set-associative cache structure

- Duplicate copies of the L1 data cache directories for coherency support
- Hardware pre-fetch support
- ECC support

1.2.2 Performance Monitoring

The performance monitoring unit (part of MPCore non-CPU logic) provides six counters, each of which can count any of the events in the processor. The unit gathers various statistics on the operation of the processor and memory system during runtime, based on ARM PMUv3 architecture.

1.3 High-Definition Audio-Video Subsystem

The audio-video subsystem off-loads audio and video processing activities from the CPU subsystem resulting in faster, fully concurrent, highly efficient operation.

1.3.1 Multi-Standard Video Decoder

The video decoder accelerates video decode, supporting low resolution content, Standard Definition (SD), High Definition (HD) and UltraHD (2160p, or 4k video) profiles. The video decoder is designed to be extremely power efficient without sacrificing performance.

The video decoder communicates with the memory controller through the video DMA which supports a variety of memory format output options. For low power operations, the video decoder can operate at the lowest possible frequency while maintaining real-time decoding using dynamic frequency scaling techniques.

Video standards supported:

- H.265: Main10, Main
- WEBM VP9 and VP8
- H.264: Baseline (no FMO/ASO support), Main, High, Stereo SEI (half-res)
- VC-1: Simple, Main, Advanced
- MPEG-4: Simple (with B frames, interlaced; no DP and RVLC)
- H.263: Profile 0
- DiVX: 4/5/6
- XviD Home Theater
- MPEG-2: MP

1.3.2 Multi-Standard Video Encoder

The multi-standard video encoder enables full hardware acceleration of various encoding standards. It performs high-quality video encoding operations for applications such as video recording and video conferencing. The encode processor is designed to be extremely power-efficient without sacrificing performance.

Video standards supported:

- H.265 Main Profile: I-frames and P-frames (No B-frames)
- H.264 Baseline/Main/High Profiles: IDR/I/P/B-frame support, MVC
- VP8
- MPEG4 (ME only)
- MPEG2 (ME only)
- VC1 (ME only): No B frame, no interlaced

1.3.3 JPEG Processing Block

The JPEG processing block is responsible for JPEG (de)compression calculations (based on JPEG still image standard), image scaling, decoding (YUV420, YUV422H/V, YUV444, YUV400) and color space conversion (RGB to YUV; decode only).

Input (encode) formats:

- Pixel width: 8bpc
- Subsample format: YUV420
- Resolution up to 16K x 16K
- Pixel pack format
  - Semi-planar/planar for 420

Output (decode) formats:

- Pixel width 8bpc
- Resolution up to 16K x 16K
- Pixel pack format
  - Semi-planar/planar for YUV420
  - YUY2/planar for 422H/422V
  - Planar for YUV444
  - Interleave for RGBA

1.3.4 Video Image Compositor (VIC)

The Video Image Compositor implements various 2D image and video operations in a power-efficient manner. It handles various system UI scaling, blending and rotation operations, video post-processing functions needed during video playback, and advanced de-noising functions used for camera capture.

Features:

- Color Decompression
- High-quality Deinterlacing
- Inverse Teleciné
- Temporal Noise Reduction
  - High-quality video playback
  - Reduces camera sensor noise
- Scaling
- Color Conversion
- Memory Format Conversion
- Blend/Composite
- 2D Bit BLIT operation
- Rotation

1.4 Image Signal Processor (ISP)

The ISP module takes data from the VI/CSI module or memory in raw Bayer format and processes it to YUV output. The imaging subsystem supports raw (Bayer) image sensors up to 24 million pixels. Advanced image processing is used to convert input to YUV data and remove artifacts introduced by high megapixel CMOS sensors and optics with up to 30-degree CRA.

Features:

- Flexible post-processing architecture for supporting custom computer vision and computational imaging operations
- Bayer domain hardware noise reduction
- Per-channel black-level compensation
- High-order lens-shading compensation
- 3 x 3 color transform
- Bad pixel correction
- Programmable coefficients for de-mosaic with color artifact reduction

  Color Artifact Reduction: a two-level (horizontal and vertical) low-pass filtering scheme that is used to reduce/remove any color artifacts that may result from Bayer signal processing and the effects of sampling an image.
- Enhanced down scaling quality
- Edge Enhancement
- Color and gamma correction
- Programmable transfer function curve
- Color-space conversion (RGB to YUV)
- Image statistics gathering (per-channel)
  - Two 256-bin image histograms
  - Up to 4,096 local region averages
  - AC flicker detection (50Hz and 60Hz)
  - Focus metric block

1.5 Display Controller Complex

The Display Controller Complex integrates two independent display controllers. Each display controller is capable of interfacing to an external display device and can drive the same or different display contents at different resolutions and refresh rates. Each controller supports a cursor and three windows (Window A, B, and C); controller A supports two additional simple windows (Window D, T). The display controller reads rendered graphics or video frame buffers in memory, blends them and sends them to the display.

Features:

- Two heads. Each can be mapped to one of:
  - 1x DSI, 1x eDP/DP (Limited Functionality: No Audio)
  - 1x HDMI/DP (Full Functionality)
- 90, 180, 270-degree image transformation uses both horizontal and vertical flips (controller A only)
- Byte-swapping options on 16-bit and 32-bit boundary for all color depths
- NVIDIA Pixel Rendering Intensity and Saturation Management™ (PRISM)
- 256 x 256 cursor size
- Color Management Unit for color decompression and to enhance color accuracy (compensate for the color error specific to the display panel being used)

- Scaling and tiling in hardware for lower power operation
- Full color alpha-blending
- Captive panels
  - Secure window (Win T) for TrustZone
  - Supports cursor and up to four windows (Win A, B, C, and D)
  - 1x 2-lane MIPI DSI
  - Supports MIPI D-PHY rates up to 1.5Gbps
  - 4-lane eDP with AUX channel
  - Independent resolution and pixel clock
  - Supports display rotation and scaling in hardware
- External displays
  - Supports cursor and three windows (Window A, B, and C)
  - 1x HDMI (2.0) or DisplayPort (HBR2) interface
  - Supports display scaling in hardware

1.6 Memory

The Jetson Nano integrates 4GB of LPDDR4 over a four-channel x 16-bit interface. Memory frequency options are 204MHz and 1600MHz; maximum frequency of 1600MHz has a theoretical peak memory bandwidth of 25.6GB/s.

The Memory Controller (MC) maximizes memory utilization while providing minimum latency access for critical CPU requests. An arbiter is used to prioritize requests, optimizing memory access efficiency and utilization and minimizing system power consumption. The MC provides access to main memory for all internal devices. It provides an abstract view of memory to its clients via standardized interfaces, allowing the clients to ignore details of the memory hierarchy. It optimizes access to shared memory resources, balancing latency and efficiency to provide best system performance, based on programmable parameters.

Features:

- TrustZone (TZ) Secure and OS-protection regions
- System Memory Management Unit
- Dual CKE signals for dynamic power down per device
- Dynamic Entry/Exit from Self -Refresh and Power Down states

The MC can sustain high utilization over a very diverse mix of requests. For example, the MC is prioritized for bandwidth (BW) over latency for all multimedia blocks (the multimedia blocks have been architected to prefetch and pipeline their operations to increase latency tolerance); this enables the MC to optimize performance by coalescing, reordering, and grouping requests to minimize memory power. DRAM also has modes for saving power when it is either not being used, or during periods of specific types of use.

2.0 Power and System Management

The Jetson Nano module operates from a single power source (VDD_IN) with all internal module voltages and I/O voltages generated from this input. This enables the on-board power management controller to implement a tiered structure of power and clock gating in a complex environment that optimizes power consumption based on workload:

- **Power Management Controller (PMC) and Real Time Clock (RTC):** These blocks reside in an Always On (not power gated) partition. The PMC provides an interface to an external power manager IC or PMU. It primarily controls voltage transitions for the SoC as it transitions to/from different low power modes; it also acts as a slave receiving dedicated power/clock request signals as well as wake events from various sources (e.g., SPI, I2C, RTC, USB attach) which can wake the system from a deep-sleep state. The RTC maintains the ability to wake the system based on either a timer event or an external trigger (e.g., key press).
- **Power Gating:** The SoC aggressively employs power-gating (controlled by PMC) to power-off modules which are idle. CPU cores are on a separate power rail to allow complete removal of power and eliminate leakage. Each CPU can be power gated independently. Software provides context save/restore to/from DRAM.
- **Clock Gating:** Used to reduce dynamic power in a variety of power states.
- **Dynamic Voltage and Frequency Scaling (DVFS):** Raises voltages and clock frequencies when demand requires, lowers them when less is sufficient, and removes them when none is needed. DVFS is used to change the voltage and frequencies in the following power domains: CPU, CORE, and GPU.

Table 1 Power and System Control Pin Descriptions

Pin	Name	Direction	Type	PoR	Description
251 252 253 254 255 256 257 258 259 260	VDD_IN	Input	5.0V		Power: Main DC input, supplies PMIC and other regulators
235	PMIC_BBAT	Bidirectional	1.65V-5.5V		Power: PMIC Battery Back-up. Optionally used to provide back-up power for the Real-Time Clock (RTC).
240	SLEEP/WAKE*	Input	CMOS – 5.0V	PU	Sleep / Wake. Configured as GPIO for optional use to place system in sleep mode or wake system from sleep.
214	FORCE_RECOVERY*	Input	CMOS – 1.8V	PU	Force Recovery: strap pin
237	POWER_EN	Input	CMOS – 5.0V		Module on/off: high = on, low = off.
233	SHUTDOWN_REQ*	Output	CMOS – 5.0V	z	Shutdown Request: used by the module to request a shutdown from the carrier board (POWER_EN low). 100kΩ pull-up to VDD_IN (5V) on the module.
239	SYS_RESET*	Bidirectional	Open Drain, 1.8V	1	Module Reset. Reset to the module when driven low by the carrier board. When module power sequence is complete used as carrier board supply enable. Used to ensure proper power on/off sequencing between module and carrier board supplies. 4.7kΩ pull-up to 1.8V on the module.
178	MOD_SLEEP*	Output	CMOS – 1.8V		Indicates the module sleep status. Low is in sleep mode, high is normal operation. This pin is controlled by system software and should not be modified.

2.1 Power Rails

VDD_IN must be supplied by the carrier board that the Jetson Nano is designed to connect to. It must meet the required electrical specifications detailed in Section 5. All Jetson Nano interfaces are referenced to on-module voltage rails; no I/O voltage is required to be supplied to the module. See the *Jetson Nano Product Design Guide* for details of connecting to each of the interfaces.

2.2 PMIC_BBAT

An optional back up battery can be attached to the PMIC_BBAT module input. It is used to maintain the RTC voltage when VDD_IN is not present. This pin is connected directly to the onboard PMIC. When a backup cell is connected to the PMIC, the RTC will retain its contents and can be configured to charge the backup cell. RTC accuracy is 2 seconds/day under typical room temperature conditions (only).

The following backup cells may be attached to the PMIC_BBAT pin:

- Super Capacitor (gold cap, double layer electrolytic)
- Standard capacitors (tantalum)
- Rechargeable Lithium Manganese cells

A backup cell **MUST** provide a voltage in the range 2.5V to 3.5V. The backup cell is charged with a constant current, constant voltage charger that can be configured between 2.5V and 3.5V (constant voltage) output and 50µA to 800µA (constant current).

Table 2: PMIC_BBAT Pin Descriptions

Pin	Name	Description	Direction	Pin Type
235	PMIC_BBAT	PMIC Battery Back-up. Optionally used to provide back-up power for the Real-Time Clock (RTC). Connects to Lithium Cell or super capacitor on Carrier Board. PMIC is supply when charging cap or coin cell. Super cap or coin cell is source when system is disconnected from power. Constant current of 2.0µA for 2.5V; 2.3µA for 3.3V typical; 4.2µA maximum.	Bidir	1.65V-5.5V

2.3 Power Domains/Islands

Power domains and power islands are used to optimize power consumption for various low-power use cases and limiting leakage current. The RTC domain is always on, CORE/CPU/GPU domains can be turned on and off. The CPU, CORE and GPU power domains also contain power-gated islands which are used to power individual modules (as needed) within each domain. Clock-gating is additionally applied during powered-on but idle periods to further reduce unnecessary power consumption. Clock-gating can be applied to both power-gated and non-power-gated islands (NPG).

Table 3 Power Domains

Power Domain	Power Island in Domain	Modules in Power Island
RTC (VDD_RTC)	N/A	PMC (Power Management Controller)
		RTC (Real Time Clock)
CORE (VDD_SOC)	NPG (Non-Power-Gated)	AHB, APB Bus, AVP, Memory Controller (MC/EMC), USB 2.0, SDMMC
	VE, VE2	ISPs (image signal processing) A and B, VI (video input), CSI (Camera Serial Interface)
	NVENC	Video Encode
	NVDEC	Video Decode
	NVJPG	JPG accelerator and additional Video Decode
	PCX	PCIe

Power Domain	Power Island in Domain	Modules in Power Island
	SOR	HDMI, DSI, DP
	IRAM	IRAM
	DISP-A, DISP-B	Display Controllers A and B
	XUSBA, XUSBB, XUSBC	USB 3.0
	VIC	VIC (Video Image Compositor)
	ADSP	APE (Audio Processing Engine)
	DFD	Debug logic
GPU (VDD_GPU)	GPU	3D, FE, PD, PE, RAST, SM, ROP
CPU (VDD_CPU)	CPU 0	CPU 0
	CPU 1	CPU 1
	CPU 2	CPU 2
	CPU 3	CPU 3
	Non-CPU	L2 Cache for Main CPU complex
	TOP	Top level logic

2.4 Power Management Controller (PMC)

The PMC power management features enable both high-speed operation and very low-power standby states. The PMC primarily controls voltage transitions for the SoC as it transitions to/from different low-power modes; it also acts as a slave receiving dedicated power/clock request signals as well as wake events from various sources (e.g., SPI, I2C, RTC, USB attach) which can wake the system from deep sleep state. The PMC enables aggressive power-gating capabilities on idle modules and integrates specific logic to maintain defined states and control power domains during sleep and deep sleep modes.

2.4.1 Resets

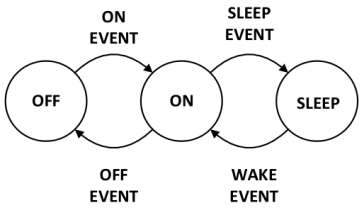
The PMC receives the primary reset event (from SYS_RESET*) and generates various resets for: PMC, RTC, and CAR. From the PMC provided reset, the Clock and Reset (CAR) controller generates resets for most of the blocks in the module. In addition to reset events, the PMC receives other events (e.g., thermal, WatchDog Timer (WDT), software, wake) which also result in variants of system reset.

The RTC block includes an embedded real-time clock and can wake the system based on either a timer event or an external trigger (e.g., key press).

2.4.2 System Power States and Transitions

The Jetson module operates in three main power modes: OFF, ON, and SLEEP. The module transitions between these states are based on various events from hardware or software. Figure 1 shows the transitions between these states.

Figure 1 Power State Diagram



2.4.2.1 ON State

The ON power state is entered from either OFF or SLEEP states. In this state the Jetson module is fully functional and operates normally. An ON event has to occur for a transition between OFF and ON states. The only ON EVENT currently used is a low to high transition on the POWER_EN pin. This must occur with VDD_IN connected to a power rail, and POWER_EN is asserted (at a logic1). The POWER_EN control is the carrier board indication to the Jetson module that the VDD_VIN power is good. The Carrier board should assert this high only when VDD_IN has reached its required voltage level and is stable. This prevents the Jetson module from powering up until the VDD_IN power is stable.

NOTE: The Jetson Nano module does include an Auto-Power-On option; a system input that enables the module to power on if asserted. For more information on available signals and broader system usage, see the *Jetson Nano Product Design Guide*.

2.4.2.2 OFF State

The OFF state is the default state when the system is not powered. It can only be entered from the ON state, through an OFF event. OFF Events are listed in the table below.

Table 4 OFF State Events

Event	Details	Preconditions
HW Shutdown	Set POWER_EN pin to zero for at least 100µS, the internal PMIC will start shutdown sequence	In ON State
SW Shutdown	Software initiated shutdown	ON state, Software operational
Thermal Shutdown	If the internal temperature of the Jetson module reaches an unsafe temperature, the hardware is designed to initiate a shutdown	Any power state

2.4.2.3 SLEEP State

The Sleep state can only be entered from the ON state. This state allows the Jetson module to quickly resume to an operational state without performing a full boot sequence. In this state the Jetson module operates in low power with enough circuitry powered to allow the device to resume and re-enter the ON state. During this state the output signals from Jetson module are maintained at their logic level prior to entering the state (i.e., they do not change to a 0V level).

The SLEEP state can only be entered directly by software. For example, operating within an OS, with no operations active for a certain time can trigger the OS to initiate a transition to the SLEEP state.

To Exit the SLEEP state a WAKE event must occur. WAKE events can occur from within the Jetson module or from external devices through various pins on the Jetson Nano connector. A full list of Wake enabled pins is available in the pinmux.

Table 5 SLEEP State Events

Event	Details
RTC WAKE up	Timers within the Jetson module can be programmed, on SLEEP entry. When these expire they create a WAKE event to exit the SLEEP state.
Thermal Condition	If the Jetson module internal temperature exceeds programmed hot and cold limits the system is forced to wake up, so it can report and take appropriate action (shut down for example)
USB VBUS detection	If VBUS is applied to the system (USB cable attached) then the device can be configured to Wake and enumerate

2.5 Thermal and Power Monitoring

The Jetson Nano is designed to operate under various workloads and environmental conditions. It has been designed so that an active or passive heat sinking solution can be attached. The module contains various methods through hardware and software to limit the internal temperature to within operating limits. See the *Jetson Nano Thermal Design Guide* for more details.

2.6 Power Sequencing

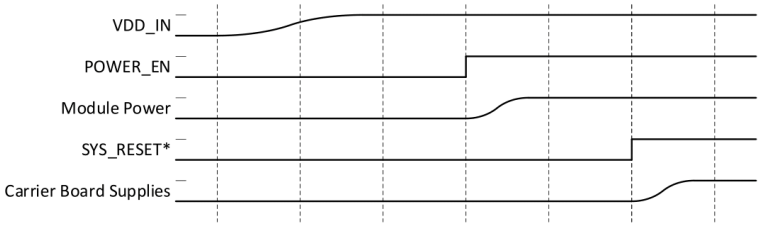
The Jetson Nano module is required to be powered on and off in a known sequence. Sequencing is determined through a set of control signals; the SYS_RESET* signal (when deasserted) is used to indicate when the carrier board can power on. The following sections provide an overview of the power sequencing steps between the carrier board and Jetson Nano module. Refer to the *Jetson Nano Product Design Guide* for system level details on the application of power, power sequencing, and monitoring. The Jetson Nano module and the product carrier board must be power sequenced properly to avoid potential damage to components on either the module or the carrier board system.

2.6.1 Power Up

During power up, the carrier board must wait until the signal SYS_RESET* is deasserted from the Jetson module before enabling its power; the Jetson module will deassert the SYS_RESET* signal to enable the complete system to boot.

> **NOTE:** I/O pins cannot be high (>0.5V) before SYS_RESET* goes high. When SYS_RESET* is low, the maximum voltage applied to any I/O pin is 0.5V. For more information, refer to the *Jetson Nano Product Design Guide*.

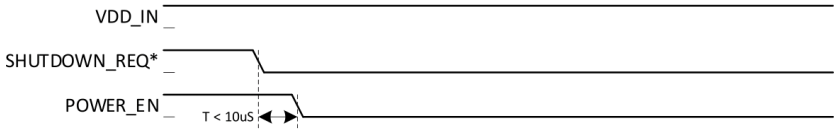
Figure 2 Power-up Sequence (No Power Button – Auto-Power-On Enabled)



2.6.2 Power Down

In a shutdown event the Jetson module asserts SHUTDOWN_REQ*. The SHUTDOWN_REQ* must be serviced by the carrier board to toggle POWER_EN from high to low, even in cases of sudden power loss. The Jetson module starts the power off sequence when POWER_EN is deasserted; SYS_RESET* is asserted by the Jetson module, allowing the carrier board to put any components into a known state and power down.

Figure 3 Power Down Sequence (Initiated by SHUTDOWN_REQ* Assertion)



3.0 Pin Descriptions

The primary interface to Jetson Nano is via a 260-pin SO-DIMM connector. Connector exposes power, ground, high-speed and low-speed industry standard I/O connections. See the *NVIDIA Jetson Nano Product Design Guide* for details on integrating the module and mating connector into product designs.

The I/O pins on the SO-DIMM are comprised of both Single Function I/O (SFIO) and Multi-Purpose digital I/O (MPIO) pins. Each MPIO can be configured to act as a GPIO or it can be assigned for use by a particular I/O controller. Though each MPIO has up to five functions (GPIO function and up to four SFIO functions), a given MPIO can only act as a single function at a given point in time. The functions for each pin on the Jetson module are fixed to a single SFIO function or as a GPIO. The different MPIO pins share a similar structure, but there are several varieties of such pins. The varieties are designed to minimize the number of on-board components (such as level shifters or pull-up resistors) required in Jetson Nano designs.

MPIO pin types:

- ST (standard) pins are the most common pins on the chip. They are used for typical General Purpose I/O.
- DD (dual-driver) pins are similar to the ST pins. A DD pin can tolerate its I/O pin being pulled up to 3.3V (regardless of supply voltage) if the pin's output-driver is set to open-drain mode. There are special power-sequencing considerations when using this functionality.

  NOTE: The output of DD pins cannot be pulled High during deep-power-down (DPD).
- CZ (controlled output impedance) pins are optimized for use in applications requiring tightly controlled output impedance. They are similar to ST pins except for changes in the drive strength circuitry and in the weak pull-ups/-downs. CZ pins are included on the VDDIO_SDMMC3 (Module SDMMC pins) power rail; also includes a CZ_COMP pin. Circuitry within the Jetson module continually matches the output impedance of the CZ pins to the on-board pull-up/-down resistors attached to the CZ_COMP pins.
- LV_CZ (low voltage-controlled impedance) pins are similar to CZ pins but are optimized for use with a 1.2V supply voltage (and signaling level). They support a 1.8V supply voltage (and signaling level) as a secondary mode. The Jetson nano uses LV_CZ pins for SPI interfaces operating at 1.8V.
- DP_AUX pin is used as an Auxiliary control channel for the DisplayPort which needs differential signaling. Because the same I/O block is used for DisplayPort and HDMI to ensure the control path to the display interface is minimized, the DP_AUX pins can operate in open-drain mode so that HDMI's control path (i.e., DDC interface which needs I2C) can also be used in the same pin.

Each MPIO pin consists of:

- An output driver with tristate capability, drive strength controls and push-pull mode, open-drain mode, or both
- An input receiver with either Schmitt mode, CMOS mode, or both
- A weak pull-up and a weak pull-down

MPIO pins are partitioned into multiple "pin control groups" with controls being configured for the group. During normal operation, these per-pin controls are driven by the pinmux controller registers. During deep sleep, the PMC bypasses and then resets the pinmux controller registers. Software reprograms these registers as necessary after returning from deep sleep.

Refer to the *Tegra X1 (SoC) Technical Reference Manual* for more information on modifying pin controls.

3.1 MPIO Power-on Reset Behavior

Each MPIO pin has a deterministic power-on reset (PoR) state. The particular reset state for each pin is chosen to minimize the need of on-board components like pull-up resistors in a Jetson Nano-based system. For example, the on-chip weak pull-ups are enabled during PoR for pins which are usually used to drive active-low chip selects.

3.2 MPIO Deep Sleep Behavior

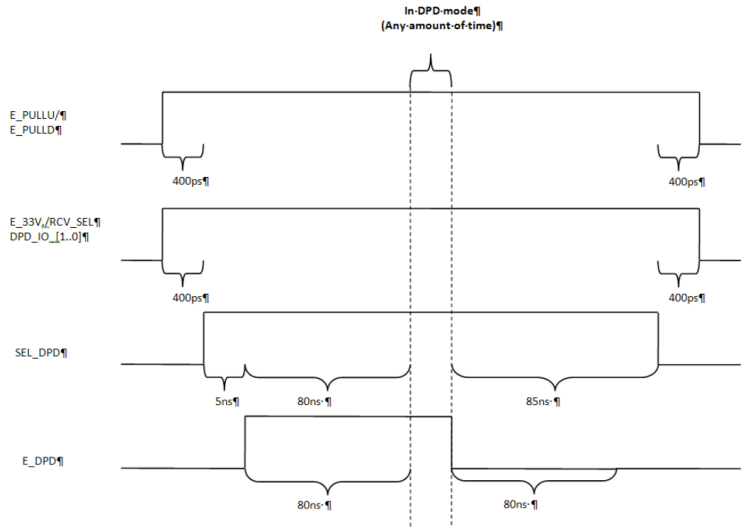
Deep Sleep is an ultra-low-power standby state in which the Jetson Nano maintains much of its I/O state while most of the chip is powered off. The following lists offer a simplified description of the deep sleep entry and exit concentrating on those aspects which relate to the MPIO pins. During deep sleep most of the pins are put in a state called Deep Power Down (DPD). The sequence for entering to DPD is same across pins. Specific variations are there in some pins in terms of type of features that are available in DPD.

> **NOTE:** The output of DD pins cannot be pulled High during deep-power-down (DPD).
> OD pins do NOT retain their output during DPD. OD pins should NOT be configured as GPIOs in a platform where they are expected to hold a value during DPD.

ALL MPIO pins **do NOT** have identical behavior during deep sleep. They differ with regard to:

- Input buffer behavior during deep sleep
  - Forcibly disabled OR
  - Enabled for use as a “GPIO wake event” OR
  - Enabled for some other purpose (e.g., a “clock request” pin)
- Output buffer behavior during deep sleep
  - Maintain a static programmable (0, 1, or tristate) constant value OR
  - Capable of changing state (i.e., dynamic while the chip is still in deep sleep)
- Weak pull-up/pull-down behavior during deep sleep
  - Forcibly disabled OR
  - Can be configured
- Pins that do not enter deep sleep
  - Some of the pins whose outputs are dynamic during deep sleep are of special type and they do not enter deep sleep (e.g., pins that are associated with PMC logic do not enter deep sleep, pins that are associated with JTAG do not enter into deep sleep any time.

Figure 4 DPD Wait Times



3.3 GPIO Pins

The Jetson Nano has multiple dedicated GPIOs. Each GPIO can be individually configurable as an Output, Input, or Interrupt source with level/edge controls. The pins listed in the following table are dedicated GPIOs; some with alternate SFIO functionality. Many other pins not included in this list are capable of being configured as GPIOs instead of the SFIO functionality the pin name suggests (e.g., UART, SPI, I²S, etc.). All pins that can support GPIO functionality have this exposed in the Pinmux.

Table 6 Dedicated GPIO Pin Descriptions

Pin	Name	Direction	Type	PoR	Alternate Function
87	GPIO00	Bidirectional	Open-Drain [DD]	0	USB VBUS Enable (USB_VBUS_EN0)
118	GPIO01	Bidirectional	CMOS – 1.8V [ST]	pd	Camera MCLK #2 (CLK)
124	GPIO02	Bidirectional	CMOS – 1.8V [ST]	pd	
126	GPIO03	Bidirectional	CMOS – 1.8V [ST]	pd	
127	GPIO04	Bidirectional	CMOS – 1.8V [ST]	pd	
128	GPIO05	Bidirectional	CMOS – 1.8V [ST]	pd	
130	GPIO06	Bidirectional	CMOS – 1.8V [ST]	pd	
206	GPIO07	Bidirectional	CMOS – 1.8V [ST]	pd	Pulse Width Modulation Signal (PWM)
208	GPIO08	Bidirectional	CMOS – 1.8V [ST]	pd	Fan Tachometer
211	GPIO09	Bidirectional	CMOS – 1.8V [ST]	pd	Audio Clock (AUD_MCLK)
212	GPIO10	Bidirectional	CMOS – 1.8V [ST]	pd	
216	GPIO11	Bidirectional	CMOS – 1.8V [ST]	pd	Camera MCLK #3
218	GPIO12	Bidirectional	CMOS – 1.8V [ST]	pd	
228	GPIO13	Bidirectional	CMOS – 1.8V [ST]	pd	Pulse Width Modulation Signal
230	GPIO14	Bidirectional	CMOS – 1.8V [ST]	pd	Pulse Width Modulation Signal
114	CAM0_PWDN	Bidirectional	CMOS – 1.8V [ST]	pd	
120	CAM1_PWDN	Bidirectional	CMOS – 1.8V [ST]	pd	

4.0 Interface Descriptions

The following sections outline the interfaces available on the Jetson Nano module and details the module pins used to interact with and control each interface. See the *Tegra X1 Series SoC Technical Reference Manual* for complete functional descriptions, programming guidelines and register listings for each of these blocks.

4.1 USB

Standard	Notes
Universal Serial Bus Specification Revision 3.0	Refer to specification for related interface timing details.
Universal Serial Bus Specification Revision 2.0	USB Battery Charging Specification, version 1.0; including Data Contact Detect protocol Modes: Host and Device Speeds: Low, Full, and High Refer to specification for related interface timing details.
Enhanced Host Controller Interface Specification for Universal Serial Bus revision 1.0	Refer to specification for related interface timing details.

An xHCI/Device controller (named XUSB) supports the xHCI programming model for scheduling transactions and interface managements as a host that natively supports USB 3.0, USB 2.0, and USB 1.1 transactions with its USB 3.0 and USB 2.0 interfaces. The XUSB controller supports USB 2.0 L1 and L2 (suspend) link power management and USB 3.0 U1, U2, and U3 (suspend) link power managements. The XUSB controller supports remote wakeup, wake on connect, wake on disconnect, and wake on overcurrent in all power states, including sleep mode.

USB 2.0 Ports

Each USB 2.0 port operates in USB 2.0 High Speed mode when connecting directly to a USB 2.0 peripheral and operates in USB 1.1 Full- and Low-Speed modes when connecting directly to a USB 1.1 peripheral. All USB 2.0 ports operating in High Speed mode share one High-Speed Bus Instance, which means 480 Mb/s theoretical bandwidth is distributed across these ports. All USB 2.0 ports operating in Full- or Low-Speed modes share one Full/Low-Speed Bus Instance, which means 12 Mb/s theoretical bandwidth is distributed across these ports.

USB 3.0 Port

The USB 3.0 port only operates in USB 3.0 Super Speed mode (5 Gb/s theoretical bandwidth).

Table 7 USB 2.0 Pin Descriptions

Pin	Name	Direction	Type	Description
87	GPIO0	Input	USB VBUS, 5V	USB 0 VBUS Detect (USB_VBUS_EN0). Do not feed 5V directly into this pin; see the *Jetson Nano Product Design Guide* for complete details.
109 111	USB0_D_N USB0_D_P	Bidirectional	USB PHY	USB 2.0 Port 0 Data
115 117	USB1_D_N USB1_D_P	Bidirectional	USB PHY	USB 2.0 Port 1 Data
121 123	USB2_D_N USB2_D_P	Bidirectional	USB PHY	USB 2.0 Port 2 Data

Table 8 USB 3.0 Pin Descriptions

Pin	Name	Direction	Type	Description
163 161	USBSS_RX_P USBSS_RX_N	Input	USB SS PHY	USB 3.0 SS Receive

Pin	Name	Direction	Type	Description
168 166	USBSS_TX_P USBSS_TX_N	Output	USB SS PHY	USB 3.0 SS Transmit

4.2 PCI Express (PCIe)

Standard	Notes
PCI Express Base Specification Revision 2.0	Jetson Nano meets the timing requirements for the Gen2 (5.0 GT/s) data rates. Refer to specification for complete interface timing details. Although NVIDIA validates that the Jetson Nano design complies with the PCIe specification, PCIe software support may be limited.

The Jetson module integrates a single PCIe Gen2 controller supporting:

- Connections to a single (x1/2/4) endpoint
- Upstream and downstream AXI interfaces that serve as the control path from the Jetson Nano to the external PCIe device.
- Gen1 (2.5 GT/s/lane) and Gen2 (5.0 GT/s/lane) speeds.

NOTE: Upstream Type 1 Vendor Defined Messages (VDM) should be sent by the Endpoint Port (EP) if the Root Port (RP) also belongs to same vendor/partner; otherwise the VDM is silently discarded.

See the *Jetson Nano Product Design Guide* for supported USB 3.0/PCIe configuration and connection examples.

Table 9 PCIe Pin Descriptions

Pin	Name	Direction	Type	PoR	Description
179	PCIE_WAKE*	Input	Open Drain 3.3V	z	PCI Express Wake This signal is used as the PCI Express defined WAKE# signal. When asserted by a PCI Express device, it is a request that system power be restored. No interrupt or other consequences result from the assertion of this signal. On module 100kΩ pull-up to 3.3V
160 162	PCIE0_CLK_N PCIE0_CLK_P	Output	PCIe PHY	0 0	PCIe Reference Clock
180	PCIE0_CLKREQ*	Bidirectional	Open Drain 3.3V	z	PCIe Reference Clock Request This signal is used by a PCIe device to indicate it needs the PCIE0_CLK_N and PCIE0_CLK_P to actively drive reference clock. On module 47kΩ pull-up to 3.3V
181	PCIE0_RST*	Output	Open Drain 3.3V	0	PCIe Reset This signal provides a reset signal to all PCIe links. It must be asserted 100 ms after the power to the PCIe slots has stabilized. On module 47kΩ pull-up to 3.3V
157 155	PCIE0_RX3_P PCIE0_RX3_N	Input	PCIe PHY		PCIe Receive (Lane 3)
151 149	PCIE0_RX2_P PCIE0_RX2_N	Input	PCIe PHY		PCIe Receive (Lane 2)
139 137	PCIE0_RX1_P PCIE0_RX1_N	Input	PCIe PHY		PCIe Receive (Lane 1)
133 131	PCIE0_RX0_P PCIE0_RX0_N	Input	PCIe PHY		PCIe Receive (Lane 0)
156 154	PCIE0_TX3_P PCIE0_TX3_N	Output	PCIe PHY		PCIe Transmit (Lane 3)

Pin	Name	Direction	Type	PoR	Description
150 148	PCIE0_TX2_P PCIE0_TX2_N	Output	PCIe PHY		PCIe Transmit (Lane 2)
142 140	PCIE0_TX1_P PCIE0_TX1_N	Output	PCIe PHY		PCIe Transmit (Lane 1)
136 134	PCIE0_TX0_P PCIE0_TX0_N	Output	PCIe PHY		PCIe Transmit (Lane 0)

4.3 Display Interfaces

The Jetson Nano Display Controller Complex integrates a MIPI-DSI interface and Serial Output Resource (SOR) to collect pixels from the output of the display pipeline, format/encode them to desired format, and then streams to various output devices. The SOR consists of several individual resources which can be used to interface with different display devices such as HDMI, DP, or eDP.

4.3.1 MIPI Display Serial Interface (DSI)

The Display Serial Interface (DSI) is a serial bit-stream replacement for the parallel MIPI DPI and DBI display interface standards. DSI reduces package pin-count and I/O power consumption. DSI support enables both display controllers to connect to an external display(s) with a MIPI DSI receiver. The DSI transfers pixel data from the internal display controller to an external third-party LCD module.

Features:

- PHY Layer
  - Start / End of Transmission. Other out-of-band signaling
  - Per DSI interface: one Clock Lane; two Data Lanes
  - Supports link configuration – 1x 2
  - Maximum link rate 1.5Gbps as per MIPI D-PHY 1.1v version
  - Maximum 10MHz LP receive rate
- Lane Management Layer with Distributor
- Protocol Layer with Packet Constructor
- Supports MIPI DSI 1.0.1v version mandatory features
- Command Mode (One-shot) with Host and/or display controller as master
- Clocks
  - Bit Clock: Serial data stream bit-rate clock
  - Byte Clock: Lane Management Layer Byte-rate clock
  - Application Clock: Protocol Layer Byte-rate clock.
- Error Detection / Correction
  - ECC generation for packet Headers
  - Checksum generation for Long Packets
- Error recovery
- High-Speed Transmit timer
- Low-Power Receive timer
- Turnaround Acknowledge Timeout

Table 10 DSI Pin Descriptions

Pin	Name	Direction	Type	Description
76 78	DSI_CLK_N DSI_CLK_P	Output	MIPI D-PHY	Differential output clock for DSI interface
82 84	DSI_D1_N DSI_D1_P	Output	MIPI D-PHY	Differential data lanes for DSI interface.
70 72	DSI_D0_N DSI_D0_P	Bidirectional	MIPI D-PHY	Differential data lanes for DSI interface. DSI lane can read data back from the panel side in low power (LP) mode.

4.3.2 High-Definition Multimedia Interface (HDMI) and DisplayPort (DP) Interfaces

Standard	Notes
High-Definition Multimedia Interface (HDMI) Specification, version 2.0	> 340MHz pixel clock Scrambling support Clock/4 support (1/40 bit-rate clock)

The HDMI and DP interfaces share the same set of interface pins. A new transport mode was introduced in HDMI 2.0 to enable link clock frequencies greater than 340MHz and up to 600MHz. For transfer rates above 340MHz, there are two main requirements:

- All link data, including active pixel data, guard bands, data islands and control islands must be scrambled.
- The TMDS clock lane must toggle at CLK/4 instead of CLK. Below 340MHz, the clock lane toggles as normal (independent of the state of scrambling).

Features:

- HDMI
  - HDMI 2.0 mode (3.4Gbps < data rate <= 6Gbps)
  - HDMI 1.4 mode (data rate<=3.4Gbps)
  - Multi-channel audio from HDA controller, up to eight channels 192kHz 24-bit.
  - Vendor Specific Info-frame (VSI) packet transmission
  - 24-bit RGB pixel formats
  - Transition Minimized Differential Signaling (TMDS) functional up to 340MHz pixel clock rate
- DisplayPort
  - Display Port mode: interface is functional up to 540MHz pixel clock rate (i.e., 1.62GHz for RBR, 2.7GHz for HBR, and 5.4GHz for HBR2).
  - 8b/10b encoding support
  - External Dual Mode standard support
  - Audio streaming support

Table 11 HDMI Pin Descriptions

Pin	Name	Direction	Type	Description
83 81	DP1_TXD3_P DP1_TXD3_N	Differential Output	AC-Coupled on Carrier Board [DP]	DP Data lane 3 or HDMI Differential Clock. AC coupling required on carrier board. For HDMI, pull-downs (with disable) also required on carrier board.
77 75 71 69 65	DP1_TXD2_P DP1_TXD2_N DP1_TXD1_P DP1_TXD1_N DP1_TXD0_P	Differential Output	AC-Coupled on Carrier Board [DP]	HDMI Differential Data lanes 2:0. AC coupling required on carrier board. For HDMI, pull-downs (with disable) also required on carrier board. HDMI: DP1_TXD2_[P,N] = HDMI Lane 0 DP1_TXD1_[P,N] = HDMI Lane 1

Pin	Name	Direction	Type	Description
63	DP1_TXD0_N			DP1_TXD0_[P,N] = HDMI Lane 2
96	DP1_HPD	Input	CMOS – 1.8V [ST]	HDMI Hot Plug detection. Level shifter required as this pin is not 5V tolerant.
94	HDMI_CEC	Bidirectional	Open Drain, 1.8V [DD]	Consumer Electronics Control (CEC) one-wire serial bus. NVIDIA provides low level CEC APIs (read/write). These are not supported in earlier Android releases. For additional CEC support, 3rd party libraries need to be made available.
100	DP1_AUX_P	Bidirectional	Open-Drain, 1.8V (3.3V tolerant - DDC) [DP_AUX]	DDC Serial Clock for HDMI. Level shifter required; pin is not 5V tolerant.
98	DP1_AUX_N	Bidirectional	Open-Drain, 1.8V (3.3V tolerant - DDC)	DDC Serial Data. Level shifter required; pin is not 5V tolerant.

Table 12 DisplayPort on DP1 Pin Descriptions

Pin	Name	Direction	Type	Description
83 81 77 75 71 69 65 63	DP1_TXD3_P DP1_TXD3_N DP1_TXD2_P DP1_TXD2_N DP1_TXD1_P DP1_TXD1_N DP1_TXD0_P DP1_TXD0_N	Differential Output	AC-Coupled on Carrier Board [DP]	DisplayPort 1 Differential Data lanes 2:0. AC coupling required on carrier board. DP1_TXD2_[P,N] = DP Lane 2 DP1_TXD1_[P,N] = DP Lane 1 DP1_TXD0_[P,N] = DP Lane 0
96	DP1_HPD	Input	CMOS – 1.8V [ST]	DisplayPort 1 Hot Plug detection. Level shifter required and must be non-inverting.
100 98	DP1_AUX_P DP1_AUX_N	Bidirectional	Open-Drain, 1.8V [DP_AUX]	DisplayPort 1 auxiliary channels. AC coupling required on carrier board.

4.3.3 Embedded DisplayPort (eDP) Interface

Standard	Notes
Embedded DisplayPort 1.4	Supported eDP 1.4 features: ▪ Additional link rates ▪ Enhanced framing ▪ Power sequencing ▪ Reduced aux timing ▪ Reduced main voltage swing

eDP is a mixed-signal interface consisting of four differential serial output lanes and one PLL. This PLL is used to generate a high frequency bit-clock from an input pixel clock enabling the ability to handle 10-bit parallel data per lane at the pixel rate for the desired mode. Embedded DisplayPort (eDP) modes (1.6GHz for RBR, 2.16GHz, 2.43GHz, 2.7GHz for HBR, 3.42GHz, 4.32GHz and 5.4GHz for HBR2).

NOTE: eDP has been tested according to DP1.2b PHY CTS even though eDPv1.4 supports lower swing voltages and additional intermediate bit rates. This means the following nominal voltage levels (400mV, 600mV, 800mV, 1200mV) and data rates (RBR, HBR, HBR2) are tested. This interface can be tuned to drive lower voltage swings below 400mV and can be programmed to other intermediate bit rates as per the requirements of the panel and the system designer.

DisplayPort on DP0 is limited to display functionality only; no HDCP or audio support.

Table 13 eDP (or DisplayPort on DP0) Pin Descriptions

Pin	Name	Direction	Type	Description
59 57 53 51 47 45 41 39	DP0_TXD3_P DP0_TXD3_N DP0_TXD2_P DP0_TXD2_N DP0_TXD1_P DP0_TXD1_N DP0_TXD0_P DP0_TXD0_N	Differential Output	AC-Coupled on Carrier Board [DP]	DP0 Differential Data. AC coupling & pull-downs (with disable) required on carrier board. DP0_TXD3_[P,N] = DisplayPort 0 Data Lane 3 DP0_TXD2_[P,N] = DisplayPort 0 Data Lane 2 DP0_TXD1_[P,N] = DisplayPort 0 Data Lane 1 DP0_TXD0_[P,N] = DisplayPort 0 Data Lane 0
88	DP0_HPD	Input	CMOS – 1.8V [ST]	DP0 Hot Plug detection. Level shifter required as this pin is not 5V tolerant
92 90	DP0_AUX_P DP0_AUX_N	Bidirectional	AC-Coupled on Carrier Board [DP_AUX]	DP0 auxiliary channels. AC coupling required on Carrier board.

4.4 MIPI Camera Serial Interface (CSI) / VI (Video Input)

Standard
MIPI CSI 2.0 Receiver specification
MIPI D-PHY® v1.2 Physical Layer specification

The Camera Serial Interface (CSI) is based on MIPI CSI 2.0 standard specification and implements the CSI receiver which receives data from an external camera module with CSI transmitter. The Video Input (VI) block receives data from the CSI receiver and prepares it for presentation to system memory or the dedicated image signal processor (ISP) execution resources.

Features:

- Supports both x4-lane and x2-lane sensor camera configurations:
  - x4 only configuration (up to three active streams)
  - x4 + x2 configurations (up to four active streams)
- Supported input data formats:
  - RGB: RGB888, RGB666, RGB565, RGB555, RGB444
  - YUV: YUV422-8b, YUV420-8b (legacy), YUV420-8b, YUV444-8b
  - RAW: RAW6, RAW7, RAW8, RAW10, RAW12, RAW14
  - DPCM: user defined
  - User defined: JPEG8
  - Embedded: Embedded control information
- Supports single-shot mode
- Physical Interface (MIPI D-PHY) Modes of Operation
  - High Speed Mode – High-speed differential signaling up to 1.5Gbps; burst transmission for low power
  - Low Power Control – Single-ended 1.2V CMOS level; low-speed signaling for handshaking.
  - Low Power Escape – Low-speed signaling for data, used for escape command entry only.

If the two streams come from a single source, then the streams are separated using a filter indexed on different data types. In case of separation using data types, the normal data type is separated from the embedded data type.

Table 14 CSI Pin Descriptions

Pin	Name	Direction	Type	Description
10	CSI0_CLK_N	Input	MIPI D-PHY	CSI 0 Clock–
12	CSI0_CLK_P	Input	MIPI D-PHY	CSI 0 Clock+
4	CSI0_D0_N	Input	MIPI D-PHY	CSI 0 Data 0–
6	CSI0_D0_P	Input	MIPI D-PHY	CSI 0 Data 0+
16	CSI0_D1_N	Input	MIPI D-PHY	CSI 0 Data 1–
18	CSI0_D1_P	Input	MIPI D-PHY	CSI 0 Data 1+
3	CSI1_D0_N	Input	MIPI D-PHY	CSI 1 Data 0–
5	CSI1_D0_P	Input	MIPI D-PHY	CSI 1 Data 0+
15	CSI1_D1_N	Input	MIPI D-PHY	CSI 1 Data 1–
17	CSI1_D1_P	Input	MIPI D-PHY	CSI 1 Data 1+
28	CSI2_CLK_N	Input	MIPI D-PHY	CSI 2 Clock–
30	CSI2_CLK_P	Input	MIPI D-PHY	CSI 2 Clock+
22	CSI2_D0_N	Input	MIPI D-PHY	CSI 2 Data 0–
24	CSI2_D0_P	Input	MIPI D-PHY	CSI 2 Data 0+
34	CSI2_D1_N	Input	MIPI D-PHY	CSI 2 Data 1–
36	CSI2_D1_P	Input	MIPI D-PHY	CSI 2 Data 1+
27	CSI3_CLK_N	Input	MIPI D-PHY	CSI 3 Clock–
29	CSI3_CLK_P	Input	MIPI D-PHY	CSI 3 Clock+
21	CSI3_D0_N	Input	MIPI D-PHY	CSI 3 Data 0–
23	CSI3_D0_P	Input	MIPI D-PHY	CSI 3 Data 0+
33	CSI3_D1_N	Input	MIPI D-PHY	CSI 3 Data 1–
35	CSI3_D1_P	Input	MIPI D-PHY	CSI 3 Data 1+
52	CSI4_CLK_N	Input	MIPI D-PHY	CSI 4 Clock–
54	CSI4_CLK_P	Input	MIPI D-PHY	CSI 4 Clock+
46	CSI4_D0_N	Input	MIPI D-PHY	CSI 4 Data 0–
48	CSI4_D0_P	Input	MIPI D-PHY	CSI 4 Data 0+
58	CSI4_D1_N	Input	MIPI D-PHY	CSI 4 Data 1–
60	CSI4_D1_P	Input	MIPI D-PHY	CSI 4 Data 1+
40	CSI4_D2_N	Input	MIPI D-PHY	CSI 4 Data 2–
42	CSI4_D2_P	Input	MIPI D-PHY	CSI 4 Data 2+
64	CSI4_D3_N	Input	MIPI D-PHY	CSI 4 Data 3–
66	CSI4_D3_P	Input	MIPI D-PHY	CSI 4 Data 3+

Table 15 Camera Clock and Control Pin Descriptions

Pin	Name	I/O	Pin Type	PoR	Description
213	CAM_I2C_SCL	Bidirectional	Open Drain – 3.3V [DD]	z	Camera I2C Clock
215	CAM_I2C_SDA	Bidirectional	Open Drain – 3.3V [DD]	z	Camera I2C Data

Pin	Name	I/O	Pin Type	PoR	Description
116	CAM0_MCLK	Output	CMOS – 1.8V [ST]	PD	Camera 1 Reference Clock
114	CAM0_PWDN	Output	CMOS – 1.8V [ST]	PD	Camera 1 Powerdown or GPIO
122	CAM1_MCLK	Output	CMOS – 1.8V [ST]	PD	Camera 2 Reference Clock
120	CAM1_PWDN	Output	CMOS – 1.8V [ST]	PD	Camera 2 Powerdown or GPIO

4.5 SD / SDIO

Standard	Notes
SD Specifications Part A2 SD Host Controller Standard Specification Version 4.00	
SD Specifications Part 1 Physical Layer Specification Version 4.00	
SD Specifications Part E1 SDIO Specification Version 4.00	Support for SD 4.0 Specification without UHS-II
Embedded Multimedia Card (eMMC), Electrical Standard 5.1	

The SecureDigital (SD)/Embedded MultiMediaCard (eMMC) controller is used to support the on-module eMMC and a single SDIO interface made available for use with SDIO peripherals; it supports Default and High-Speed modes.

The SDMMC controller has a direct memory interface and is capable of initiating data transfers between memory and external device. The SDMMC controller supports both the SD and eMMC bus protocol and has an APB slave interface to access configuration registers. Interface is intended for supporting various compatible peripherals with an SD/MMC interface.

Table 16 SD/SDIO Controller I/O Capabilities

Controller	Bus Width	Supported Voltages (V)	I/O bus clock (MHz)	Max Bandwidth (MBps)	Notes
SD/SDIO Card	4	1.8 / 3.3	208	104	Available at connector for SDIO or SD Card use
eMMC	8	1.8	200	400	On-module eMMC

Table 17 SD/SDIO Pin Descriptions

Pin	Name	I/O	Pin Type	PoR	Description
229	SDMMC_CLK	Output	CMOS – 1.8V / 3.3V [CZ]	PD	SDIO/MMC Clock
227	SDMMC_CMD	Bidirectional	CMOS – 1.8V / 3.3V [CZ]	PU	SDIO/MMC Command
225 223 221 219	SDMMC_DAT3 SDMMC_DAT2 SDMMC_DAT1 SDMMC_DAT0	Bidirectional	CMOS – 1.8V / 3.3V [CZ]	PU	SDIO/MMC Data bus

Note: Pin voltage is determined by LDO on module setting.

4.6 Inter-IC Sound (I²S)

Standard
Inter-IC Sound (I²S) specification

The I²S controller transports streaming audio data between system memory and an audio codec. The I²S controller supports I²S format, Left-justified Mode format, Right-justified Mode format, and DSP mode format, as defined in the Philips inter-IC-sound (I²S) bus specification.

The I²S and PCM (master and slave modes) interfaces support clock rates up to 24.5760MHz.

The I²S controller supports point-to-point serial interfaces for the I²S digital audio streams. I²S-compatible products, such as compact disc players, digital audio tape devices, digital sound processors, and those with digital TV sound may be directly connected to the I²S controller. The controller also supports the PCM and telephony mode of data-transfer. Pulse-Code-Modulation (PCM) is a standard method used to digitize audio (particularly voice) patterns for transmission over digital communication channels. The Telephony mode is used to transmit and receive data to and from an external mono CODEC in a slot-based scheme of time-division multiplexing (TDM). The I²S controller supports bidirectional audio streams and can operate in half-duplex or full-duplex mode.

Features:

- Basic I²S modes to be supported (I²S, RJM, LJM and DSP) in both Master and Slave modes.
- PCM mode with short (one-bit-clock wide) and long-fsync (two bit-clocks wide) in both master and slave modes.
- NW-mode with independent slot-selection for both Tx and Rx
- TDM mode with flexibility in number of slots and slot(s) selection.
- Capability to drive-out a High-z outside the prescribed slot for transmission
- Flow control for the external input/output stream.

Table 18 Audio Pin Descriptions

Pin	Name	Direction	Type	PoR	Description
211	GPIO09	Output	CMOS – 1.8V [ST]	PD	Audio Codec Master Clock (AUD_MCLK)
195	I2S0_DIN	Input	CMOS – 1.8V [CZ]	PD	I²S Audio Port 0 Data In
193	I2S0_DOUT	Output	CMOS – 1.8V [CZ]	PD	I²S Audio Port 0 Data Out
197	I2S0_FS	Bidirectional	CMOS – 1.8V [CZ]	PD	I²S Audio Port 0 Frame Select (Left/Right Clock)
199	I2S0_SCLK	Bidirectional	CMOS – 1.8V [CZ]	PD	I²S Audio Port 0 Clock
222	I2S1_DIN	Input	CMOS – 1.8V [ST]	PD	I²S Audio Port 1 Data In
220	I2S1_DOUT	Output	CMOS – 1.8V [ST]	PD	I²S Audio Port 1 Data Out
224	I2S1_FS	Bidirectional	CMOS – 1.8V [ST]	PD	I²S Audio Port 1 Frame Select (Left/Right Clock)
226	I2S1_SCLK	Bidirectional	CMOS – 1.8V [ST]	PD	I²S Audio Port 1 Clock

4.7 Miscellaneous Interfaces

4.7.1 Inter-Chip Communication (I2C)

Standard
NXP inter-IC-bus (I^2C) specification

This general purpose I^2C controller allows system expansion for I^2C -based devices as defined in the NXP inter-IC-bus (I^2C) specification. The I^2C bus supports serial device communications to multiple devices; the I^2C controller handles clock source negotiation, speed negotiation for standard and fast devices, 7-bit slave address support according to the I^2C protocol and supports master and slave mode of operation.

The I^2C controller supports the following operating modes: Master – Standard-mode (up to 100Kbit/s), Fast-mode (up to 400 Kbit/s), Fast-mode plus (Fm+, up to 1Mbit/s); Slave – Standard-mode (up to 100Kbit/s), Fast-mode (up to 400 Kbit/s), Fast-mode plus (Fm+, up to 1Mbit/s).

Table 19 I2C Pin Descriptions

Pin	Name	I/O	Pin Type	PoR	Description
185 187	I2C0_SCL I2C0_SDA	Bidirectional	Open Drain – 3.3V [DD]	z z	Only 3.3V devices supported without level shifter. I^2C 0 Clock/Data pins. On module 2.2kΩ pull-up to 3.3V.
189 191	I2C1_SCL I2C1_SDA	Bidirectional	Open Drain – 3.3V [DD]	z z	Only 3.3V devices supported without level shifter. I^2C 1 Clock/Data pins. On module 2.2kΩ pull-up to 3.3V.
232 234	I2C2_SCL I2C2_SDA	Bidirectional	Open Drain – 1.8V [DD]	z z	Only 1.8V devices supported without level shifter. I^2C 2 Clock/Data pins. On module 2.2kΩ pull-up to 1.8V.
213 215	CAM_I2C_SCL CAM_I2C_SDA	Bidirectional	Open Drain – 3.3V [DD]	z z	Only 3.3V devices supported without level shifter. Camera I^2C Clock/Data pins. On module 4.7kΩ pull-up to 3.3V.

4.7.2 Serial Peripheral Interface (SPI)

The SPI controllers operate up to 65Mbps in master mode and 45Mbps in slave mode. It allows a duplex, synchronous, serial communication between the controller and external peripheral devices. It consists of four signals, SS_N (Chip select), SCK (clock), MOSI (Master data out and Slave data in) and MISO (Slave data out and master data in). The data is transferred on MOSI or MISO based on the data transfer direction on every SCK edge. The receiver always receives the data on the other edge of SCK.

Features:

- Independent Rx FIFO and Tx FIFO.
- Software controlled bit-length supports packet sizes of 1 to 32 bits.
- Packed mode support for bit-length of 7 (8-bit packet size) and 15 (16-bit packet size).
- SS_N can be selected to be controlled by software, or it can be generated automatically by the hardware on packet boundaries.
- Receive compare mode (controller listens for a specified pattern on the incoming data before receiving the data in the FIFO).
- Simultaneous receive and transmit supported
- Supports Master mode. Slave mode has not been validated

Table 20 SPI Pin Descriptions

Pin	Name	Direction	Type	PoR	Description
95	SPI0_CS0*	Bidirectional	CMOS – 1.8V [LV-CZ]	PU	SPI 0 Chip Select 0
97	SPI0_CS1*	Bidirectional	CMOS – 1.8V [LV-CZ]	PU	SPI 0 Chip Select 1
93	SPI0_MISO	Bidirectional	CMOS – 1.8V [LV-CZ]	PD	SPI 0 Master In / Slave Out
89	SPI0_MOSI	Bidirectional	CMOS – 1.8V [LV-CZ]	PD	SPI 0 Master Out / Slave In
91	SPI0_SCK	Bidirectional	CMOS – 1.8V [LV-CZ]	PD	SPI 0 Clock
110	SPI1_CS0*	Bidirectional	CMOS – 1.8V [CZ]	PU	SPI 1 Chip Select 0
112	SPI1_CS1*	Bidirectional	CMOS – 1.8V [CZ]	PU	SPI 1 Chip Select 1
108	SPI1_MISO	Bidirectional	CMOS – 1.8V [CZ]	PD	SPI 1 Master In / Slave Out
104	SPI1_MOSI	Bidirectional	CMOS – 1.8V [CZ]	PD	SPI 1 Master Out / Slave In
106	SPI1_SCK	Bidirectional	CMOS – 1.8V [CZ]	PD	SPI 1 Clock

Figure 5 SPI Master Timing Diagram

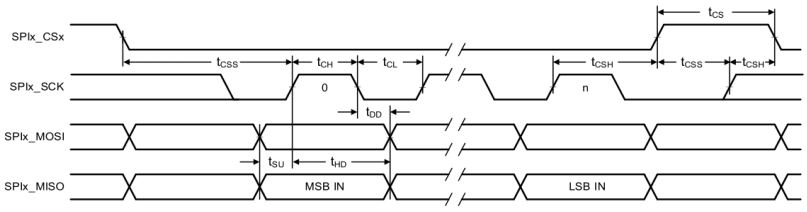


Table 21 SPI Master Timing Parameters

Symbol	Parameter	Minimum	Maximum	Unit
Fsck	SPIx_SCK clock frequency		65	MHz
Psck	SPIx_SCK period	1/Fsck		ns
t_{CH}	SPIx_SCK high time	50%Psck -10%	50%Psck +10%	ns
t_{CL}	SPIx_SCK low time	50%Psck -10%	50%Psck +10%	ns
t_{CRT}	SPIx_SCK rise time (slew rate)	0.1		V/ns
t_{CFT}	SPIx_SCK fall time (slew rate)	0.1		V/ns
t_{SU}	SPIx_MISO setup to SPIx_SCK rising edge	2		ns
t_{HD}	SPIx_MISO hold from SPIx_SCK rising edge	3		ns
t_{DD}	SPIx_MOSI delay from SPIx_SCK falling edge	0	4	ns
t_{CSS}	SPIx_CSx setup time	2		ns
t_{CSH}	SPIx_CSx hold time	3		ns
t_{CS}	SPIx_CSx high time	10		ns

Note: Polarity of SCLK is programmable. Data can be driven or input relative to either the rising edge (shown above) or falling edge.

Figure 6 SPI Slave Timing Diagram

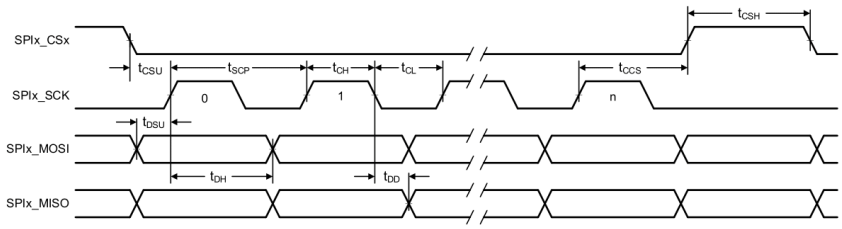


Table 22 SPI Slave Timing Parameters

Symbol	Parameter	Minimum	Maximum	Unit
t_{SCP}	SPIx_SCK period	$2*(t_{SDD}+ t_{MSU}^{1})$		ns
t_{SCH}	SPIx_SCK high time	$t_{SDD} + t_{MSU}^{1}$		ns
t_{SCL}	SPIx_SCK low time	$t_{SDD} + t_{MSU}^{1}$		ns
t_{SCSU}	SPIx_CSx setup time	1		t_{SCP}
t_{SCSH}	SPIx_CSx high time	1		t_{SCP}
t_{SCCS}	SPIx_SCK rising edge to SPIx_CSx rising edge	1	1	t_{SCP}
t_{SDSU}	SPIx_MOSI setup to SPIx_SCK rising edge	1	1	ns
t_{SDH}	SPIx_MOSI hold from SPIx_SCK rising edge	2	11	ns

1. t_{MSU} is the setup time required by the external master

Note: Polarity of SCLK is programmable. Data can be driven or input relative to either the rising edge (shown above) or falling edge.

4.7.3 UART

UART controller provides serial data synchronization and data conversion (parallel-to-serial and serial-to-parallel) for both receiver and transmitter sections. Synchronization for serial data stream is accomplished by adding start and stop bits to the transmit data to form a data character. Data integrity is accomplished by attaching a parity bit to the data character. The parity bit can be checked by the receiver for any transmission bit errors.

NOTE: The UART receiver input has low baud rate tolerance in 1-stop bit mode. External devices must use 2 stop bits.
In 1-stop bit mode, the Tegra UART receiver can lose sync between Tegra receiver and the external transmitter resulting in data errors/corruption. In 2-stop bit mode, the extra stop bit allows the Tegra UART receiver logic to align properly with the UART transmitter.

Features:

- Synchronization for the serial data stream with start and stop bits to transmit data and form a data character
- Supports both 16450- and 16550-compatible modes. Default mode is 16450
- Device clock up to 200MHz, baud rate of 12.5Mbits/second
- Data integrity by attaching parity bit to the data character
- Support for word lengths from five to eight bits, an optional parity bit and one or two stop bits
- Support for modem control inputs
- DMA capability for both Tx and Rx
- 8-bit x 36 deep Tx FIFO

- 11-bit x 36 deep Rx FIFO. Three bits of 11 bits per entry log the Rx errors in FIFO mode (break, framing, and parity errors as bits 10, 9, 8 of FIFO entry)
- Auto sense baud detection
- Timeout interrupts to indicate if the incoming stream stopped
- Priority interrupts mechanism
- Flow control support on RTS and CTS
- Internal loopback
- SIR encoding/decoding (3/16 or 4/16 baud pulse widths to transmit bit zero)

Table 23 UART Pin Descriptions

Pin	Name	Direction	Type	PoR	Description
99	UART0_TXD	Output	CMOS – 1.8V [ST]	PD	UART 0 Transmit
101	UART0_RXD	Input	CMOS – 1.8V [ST]	PU	UART 0 Receive
103	UART0_RTS*	Output	CMOS – 1.8V [ST]	PD	UART 0 Request to Send
105	UART0_CTS*	Input	CMOS – 1.8V [ST]	PD	UART 0 Clear to Send
203	UART1_TXD	Output	CMOS – 1.8V [ST]	PD	UART 1 Transmit
205	UART1_RXD	Input	CMOS – 1.8V [ST]	PD	UART 1 Receive
207	UART1_RTS*	Output	CMOS – 1.8V [ST]	PD	UART 1 Request to Send
209	UART1_CTS*	Input	CMOS – 1.8V [ST]	PD	UART 1 Clear to Send
236	UART2_TXD	Output	CMOS – 1.8V [ST]	PD	UART 2 Transmit
238	UART2_RXD	Input	CMOS – 1.8V [ST]	PD	UART 2 Receive

4.7.4 Gigabit Ethernet

The Jetson Nano integrates a Realtek RTL81119ICG Gigabit Ethernet controller. The on-module Ethernet controller supports:

- 10/100/1000 Mbps Gigabit Ethernet
- IEEE 802.3u Media Access Controller (MAC)

Table 24 Gigabit Ethernet Pin Descriptions

Pin	Name	Direction	Type	Description
194	GBE_LED_ACT	Output		Activity LED (yellow) enable
188	GBE_LED_LINK	Output		Link LED (green) enable. Link LED only illuminates if link established is 1000. 100/10 will not cause the Link LED to light up.
184 186	GBE_MDI0_N GBE_MDI0_P	Bidirectional	MDI	GbE Transformer Data 0
190 192	GBE_MDI1_N GBE_MDI1_P	Bidirectional	MDI	GbE Transformer Data 1
196 198	GBE_MDI2_N GBE_MDI2_P	Bidirectional	MDI	GbE Transformer Data 2
202 204	GBE_MDI3_N GBE_MDI3_P	Bidirectional	MDI	GbE Transformer Data 3

4.7.5 Fan

The Jetson Nano includes PWM and Tachometer functionality to enable fan control as part of a thermal solution. The Pulse Width Modulator (PWM) controller is a frequency divider with a varying pulse width. The PWM runs off a device clock programmed in the Clock and Reset controller and can be any frequency up to the device clock maximum speed of 48MHz. The PWFM gets divided by 256 before being subdivided based on a programmable value.

Table 25 Fan Pin Descriptions

Pin	Name	Direction	Type	PoR	Description
230	GPIO14	Output	CMOS – 1.8V [ST]	PD	Fan PWM
208	GPIO08	Input	CMOS – 1.8V [ST]	PD	Fan Tachometer

4.7.6 Debug

A debug interface is supported via JTAG on-module test points or serial interface over UART1. The JTAG interface can be used for SCAN testing or communicating with integrated CPU. See the *NVIDIA Jetson Nano Product Design Guide* for more information.

Table 26 Debug Pin Descriptions

Pin	Name	I/O	Pin Type	PoR	Description
-	JTAG_RTCK	Output	CMOS – 1.8V [JT_RST]	0	Return Test Clock
-	JTAG_TCK	Input	CMOS – 1.8V [JT_RST]	z	Test Clock
-	JTAG_TDI	Input	CMOS – 1.8V [JT_RST]	PU	Test Data In
-	JTAG_TDO	Output	CMOS – 1.8V [ST]	z	Test Data Out
-	JTAG_TMS	Input	CMOS – 1.8V [JT_RST]	PU	Test Mode Select
-	JTAG_GP0	Input	CMOS – 1.8V [JT_RST]	PD	Test Reset
236	UART2_TXD	Output	CMOS – 1.8V [ST]	PD	Debug UART Transmit
238	UART2_RXD	Input	CMOS – 1.8V [ST]	PD	Debug UART Receive

5.0 Physical / Electrical Characteristics

5.1 Operating and Absolute Maximum Ratings

The parameters listed in following table are specific to a temperature range and operating voltage. Operating the Jetson Nano module beyond these parameters is not recommended. Exceeding these conditions for extended periods may adversely affect device reliability.

WARNING: Exceeding the listed conditions may damage and/or affect long-term reliability of the part. The Jetson Nano module should never be subjected to conditions extending beyond the ratings listed below.

Table 27 Recommended Operating Conditions

Symbol	Parameter	Minimum	Typical	Maximum	Unit
VDD_{DC}	VDD_IN	4.75	5.0	5.25	V
	PMIC_BBAT	1.65		5.5	V

Absolute maximum ratings describe stress conditions. These parameters do not set minimum and maximum operating conditions that will be tolerated over extended periods of time. If the device is exposed to these parameters for extended periods of time, no guarantee is made and device reliability may be affected. It is not recommended to operate the Jetson Nano module under these conditions.

Table 28 Absolute Maximum Ratings

Symbol	Parameter	Minimum	Maximum	Unit	Notes
VDD_{MAX}	VDD_IN	-0.5	5.5	V	
	PMIC_BBAT	-0.3	6.0	V	
IDD_{MAX}	VDD_IN Imax		5	A	
V_{M_PIN}	Voltage applied to any powered I/O pin	-0.5	VDD + 0.5	V	VDD + 0.5V when CARRIER_PWR_ON high & associated I/O rail powered. I/O pins cannot be high (>0.5V) before CARRIER_PWR_ON goes high. When CARRIER_PWR_ON is low, the maximum voltage applied to any I/O pin is 0.5V
	DD pins configured as open drain	-0.5	3.63	V	The pin's output-driver must be set to open-drain mode
T_{OP}	Operating Temperature	-25	97	°C	See the *Jetson Nano Thermal Design Guide* for details.
T_{STG}	Storage Temperature (ambient)	-40	80	°C	
M_{MAX}	Mounting Force		4.0	kgf	kilogram-force (kgf). Maximum force applied to PCB. See the *Jetson Nano Thermal Design Guide* for additional details on mounting a thermal solution.

5.2 Digital Logic

Voltages less than the minimum stated value can be interpreted as an undefined state or logic level low which may result in unreliable operation. Voltages exceeding the maximum value can damage and/or adversely affect device reliability.

Table 29. CMOS Pin Type DC Characteristics

Symbol	Description	Minimum	Maximum	Units
V_{IL}	Input Low Voltage	-0.5	0.25 x VDD	V
V_{IH}	Input High Voltage	0.75 x VDD	0.5 + VDD	V
V_{OL}	Output Low Voltage (I_{OL} = 1mA)	---	0.15 x VDD	V
V_{OH}	Output High Voltage (I_{OH} = -1mA)	0.85 x VDD	---	V

Table 30 Open Drain Pin Type DC Characteristics

Symbol	Description	Minimum	Maximum	Units
V_{IL}	Input Low Voltage	-0.5	0.25 x VDD	V
V_{IH}	Input High Voltage	0.75 x VDD	3.63	V
V_{OL}	Output Low Voltage (I_{OL} = 1mA)	---	0.15 x VDD	V
	I2C[1,0] Output Low Voltage (I_{OL} = 2mA) (see note)	---	0.3 x VDD	V
V_{OH}	Output High Voltage (I_{OH} = -1mA)	0.85 x VDD	---	V

Note: I2C[1,0]_[SCL, SDA] pins pull-up to 3.3V through on module 2.2kΩ resistor. I2C2_[SCL, SDA] pins pull-up to 1.8V through on module 2.2kΩ resistor.

5.3 Environmental & Mechanical Screening

Module performance was assessed against a series of industry standard tests designed to evaluate robustness and estimate the failure rate of an electronic assembly in the environment in which it will be used. Mean Time Between Failures (MTBF) calculations are produced in the design phase to predict a product's future reliability in the field.

Table 31 Jetson Nano Reliability Report

Test	Reference Standards / Test Conditions
Temperature Humidity Biased	JESD22-A101 85°C / 85% RH, 168 hours, Power ON
Temperature Cycling	JESD22-A104, IPC9701 -40°C to 105°C, 250 cycles, non-operational
Humidity Steady State	NVIDIA Standard 45°C 90% RH 336hrs, operational
Mechanical Shock – 140G	JESD22-B110 140G, half sine, 1 shock/orientation, 6 orientations total, non-operational
Mechanical Shock – 50G	IEC600068-2-27 50G, half sine,1 shock/orientation, 6 orientations total, operational
Connector Insertion Cycling	EIA-364 30 cycles
Sine Vibration – 3G	IEC60068-2-6 3G, 10-500 Hz, 1 sweep/axis, 3 axes total, non-operational
Random Vibration – 2G	IEC60068-2-64 10-500 Hz, 2 Grms,1 hour/axis, non-operational

Test	Reference Standards / Test Conditions
Random Vibration – 1G	IEC60068-2-64 10-500 Hz, 1 Grms,1 hour/axis, operational
Hard Boot	NVIDIA Standard Power ON/OFF, ON for 150 sec OFF for 30 sec 1000 cycles at 25°C, 1000 cycles at -40°C
Operational Low Temp	NVIDIA Standard -5°C, 24 hours, operational
Operational High Temp	NVIDIA Standard 40°C, 90%RH, 168 hours, operational
MTBF / Failure Rate: 3,371K Hours	Telcordia SR-332, ISSUE 3 Parts Count (Method I) Controlled Environment (GB), T = 35°C, CL = 90%
MTBF / Failure Rate: 1,836K Hours	Telcordia SR-332, ISSUE 3 Parts Count (Method I) Uncontrolled Environment (GF), T = 35°C, CL = 90%
MTBF / Failure Rate: 957K Hours	Telcordia SR-332, ISSUE 3 Parts Count (Method I) Uncontrolled Environment (GM), T = 35°C, CL = 90%

5.4 Pinout

Signal Name	Pin # Top Odd	Pin # Bottom Even	Signal Name
GND	1	2	GND
CSI1_D0_N	3	4	CSI0_D0_N
CSI1_D0_P	5	6	CSI0_D0_P
GND	7	8	GND
RSVD	9	10	CSI0_CLK_N
RSVD	11	12	CSI0_CLK_P
GND	13	14	GND
CSI1_D1_N	15	16	CSI0_D1_N
CSI1_D1_P	17	18	CSI0_D1_P
GND	19	20	GND
CSI3_D0_N	21	22	CSI2_D0_N
CSI3_D0_P	23	24	CSI2_D0_P
GND	25	26	GND
CSI3_CLK_N	27	28	CSI2_CLK_N
CSI3_CLK_P	29	30	CSI2_CLK_P
GND	31	32	GND
CSI3_D1_N	33	34	CSI2_D1_N
CSI3_D1_P	35	36	CSI2_D1_P
GND	37	38	GND
DP0_TXD0_N	39	40	CSI4_D2_N
DP0_TXD0_P	41	42	CSI4_D2_P
GND	43	44	GND
DP0_TXD1_N	45	46	CSI4_D0_N
DP0_TXD1_P	47	48	CSI4_D0_P
GND	49	50	GND
DP0_TXD2_N	51	52	CSI4_CLK_N
DP0_TXD2_P	53	54	CSI4_CLK_P
GND	55	56	GND
DP0_TXD3_N	57	58	CSI4_D1_N
DP0_TXD3_P	59	60	CSI4_D1_P
GND	61	62	GND
DP1_TXD0_N	63	64	CSI4_D3_N
DP1_TXD0_P	65	66	CSI4_D3_P
GND	67	68	GND
DP1_TXD1_N	69	70	DSI_D0_N
DP1_TXD1_P	71	72	DSI_D0_P
GND	73	74	GND
DP1_TXD2_N	75	76	DSI_CLK_N
DP1_TXD2_P	77	78	DSI_CLK_P
GND	79	80	GND
DP1_TXD3_N	81	82	DSI_D1_N
DP1_TXD3_P	83	84	DSI_D1_P
GND	85	86	GND
GPIO0	87	88	DP0_HPD
SPI0_MOSI	89	90	DP0_AUX_N
SPI0_SCK	91	92	DP0_AUX_P
SPI0_MISO	93	94	HDMI_CEC
SPI0_CS0*	95	96	DP1_HPD
SPI0_CS1*	97	98	DP1_AUX_N
UART0_TXD	99	100	DP1_AUX_P
UART0_RXD	101	102	GND
UART0_RTS*	103	104	SPI1_MOSI
UART0_CTS*	105	106	SPI1_SCK
GND	107	108	SPI1_MISO
USB0_D_N	109	110	SPI1_CS0*
USB0_D_P	111	112	SPI1_CS1*
GND	113	114	CAM0_PWDN
USB1_D_N	115	116	CAM0_MCLK
USB1_D_P	117	118	GPIO01
GND	119	120	CAM1_PWDN
USB2_D_N	121	122	CAM1_MCLK
USB2_D_P	123	124	GPIO02
GND	125	126	GPIO03
GPIO04	127	128	GPIO05
GND	129	130	GPIO06
PCIE0_RX0_N	131	132	GND

Signal Name	Pin # Top Odd	Pin # Bottom Even	Signal Name
PCIE0_RX0_P	133	134	PCIE0_TX0_N
GND	135	136	PCIE0_TX0_P
PCIE0_RX1_N	137	138	GND
PCIE0_RX1_P	139	140	PCIE0_TX1_N
GND	141	142	PCIE0_TX1_P
RSVD	143	144	GND
KEY	KEY	KEY	KEY
RSVD	145	146	GND
GND	147	148	PCIE0_TX2_N
PCIE0_RX2_N	149	150	PCIE0_TX2_P
PCIE0_RX2_P	151	152	GND
GND	153	154	PCIE0_TX3_N
PCIE0_RX3_N	155	156	PCIE0_TX3_P
PCIE0_RX3_P	157	158	GND
GND	159	160	PCIE0_CLK_N
USBSS_RX_N	161	162	PCIE0_CLK_P
USBSS_RX_P	163	164	GND
GND	165	166	USBSS_TX_N
RSVD	167	168	USBSS_TX_P
RSVD	169	170	GND
GND	171	172	RSVD
RSVD	173	174	RSVD
RSVD	175	176	GND
GND	177	178	MOD_SLEEP*
PCIE_WAKE*	179	180	PCIE0_CLKREQ*
PCIE0_RST*	181	182	RSVD
RSVD	183	184	GBE_MDI0_N
I2C0_SCL	185	186	GBE_MDI0_P
I2C0_SDA	187	188	GBE_LED_LINK
I2C1_SCL	189	190	GBE_MDI1_N
I2C1_SDA	191	192	GBE_MDI1_P
I2S0_DOUT	193	194	GBE_LED_ACT
I2S0_DIN	195	196	GBE_MDI2_N
I2S0_FS	197	198	GBE_MDI2_P
I2S0_SCLK	199	200	GND
GND	201	202	GBE_MDI3_N
UART1_TXD	203	204	GBE_MDI3_P
UART1_RXD	205	206	GPIO07
UART1_RTS*	207	208	GPIO08
UART1_CTS*	209	210	CLK_32K_OUT
GPIO09	211	212	GPIO10
CAM_I2C_SCL	213	214	FORCE_RECOVERY*
CAM_I2C_SDA	215	216	GPIO11
GND	217	218	GPIO12
SDMMC_DAT0	219	220	I2S1_DOUT
SDMMC_DAT1	221	222	I2S1_DIN
SDMMC_DAT2	223	224	I2S1_FS
SDMMC_DAT3	225	226	I2S1_SCLK
SDMMC_CMD	227	228	GPIO13
SDMMC_CLK	229	230	GPIO14
GND	231	232	I2C2_SCL
SHUTDOWN_REQ*	233	234	I2C2_SDA
PMIC_BBAT	235	236	UART2_TXD
POWER_EN	237	238	UART2_RXD
SYS_RESET*	239	240	SLEEP/WAKE*
GND	241	242	GND
GND	243	244	GND
GND	245	246	GND
GND	247	248	GND
GND	249	250	GND
VDD_IN	251	252	VDD_IN
VDD_IN	253	254	VDD_IN
VDD_IN	255	256	VDD_IN
VDD_IN	257	258	VDD_IN
VDD_IN	259	260	VDD_IN

5.5 Package Drawing and Dimensions

Table 32 Module Dimensions

Description	Minimum	Typical	Maximum	Unit
Connector to opposite side			45	mm
Side (perpendicular to connector) to opposite side			69.6	mm
SoC height	1.36	1.51	1.66	mm

Figure 7 Module Top and Side View with Cover Outline

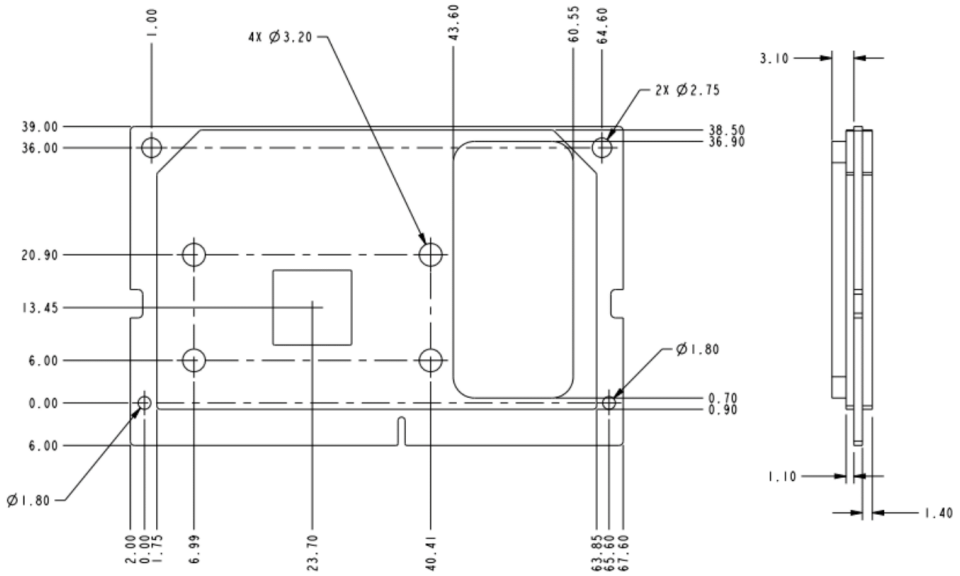


Figure 8 Module Bottom with Cover Outline

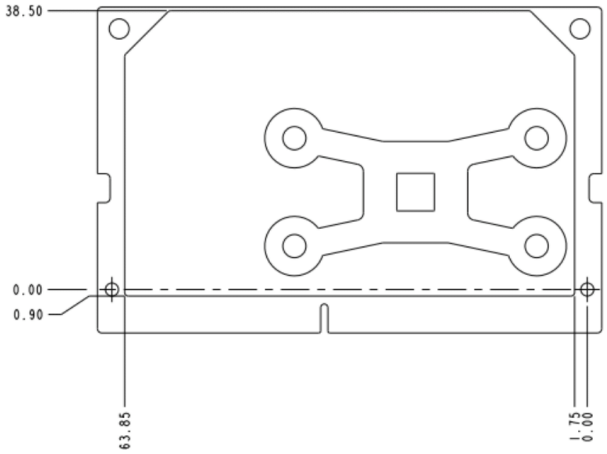


Figure 9 Module Top Showing DRAM Placement and Side View

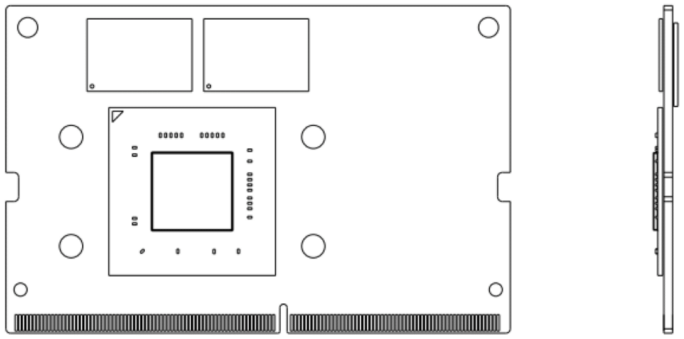
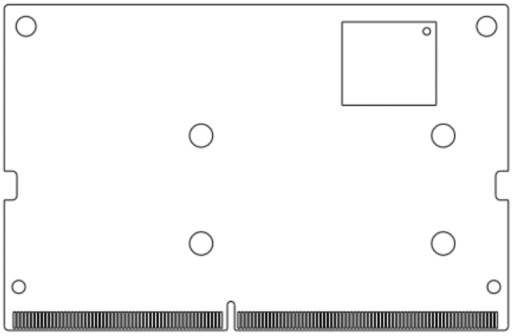


Figure 10 Module Bottom Showing EMMC Placement



Notice

The information provided in this specification is believed to be accurate and reliable as of the date provided. However, NVIDIA Corporation (“NVIDIA”) does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This publication supersedes and replaces all other specifications for the product that may have been previously supplied.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and other changes to this specification, at any time and/or to discontinue any product or service without notice. Customer should obtain the latest relevant specification before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer. NVIDIA hereby expressly objects to applying any customer general terms and conditions with regard to the purchase of the NVIDIA product referenced in this specification.

NVIDIA products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer’s own risk.

NVIDIA makes no representation or warranty that products based on these specifications will be suitable for any specified use without further testing or modification. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to ensure the product is suitable and fit for the application planned by customer and to do the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer’s product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this specification. NVIDIA does not accept any liability related to any default, damage, costs or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this specification, or (ii) customer product designs.

No license, either express or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this specification. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA. Reproduction of information in this specification is permissible only if reproduction is approved by NVIDIA in writing, is reproduced without alteration, and is accompanied by all associated conditions, limitations, and notices.

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, “MATERIALS”) ARE BEING PROVIDED “AS IS.” NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA’s aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the NVIDIA terms and conditions of sale for the product.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

ARM

ARM, AMBA and ARM Powered are registered trademarks of ARM Limited. Cortex, MPCore and Mali are trademarks of ARM Limited. All other brands or product names are the property of their respective holders. "ARM" is used to represent ARM Holdings plc; its operating company ARM Limited; and the regional subsidiaries ARM Inc.; ARM KK; ARM Korea Limited.; ARM Taiwan Limited; ARM France SAS; ARM Consulting (Shanghai) Co. Ltd.; ARM Germany GmbH; ARM Embedded Technologies Pvt. Ltd.; ARM Norway, AS and ARM Sweden AB

OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

Trademarks

NVIDIA, the NVIDIA logo, and Tegra are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

NVIDIA Corporation | 2788 San Tomas Expressway | Santa Clara, CA 95051 | www.nvidia.com

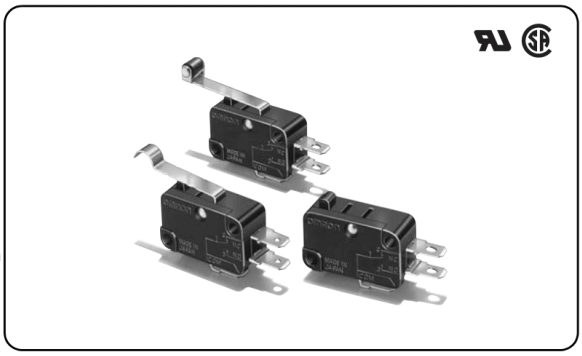
Appendix I – Omron V-156-1C25 Microswitch Datasheet

V
Miniature Basic Switch

Miniature Basic Switch that Offers High Reliability and Security

- **Wide variation of best-selling microswitches with switching currents of 10 to 21 A.**
- **Can be used for interrupting current when doors are opened or closed.**
- **Available in two types of cases: thermoplastic resin and thermosetting resin.**

RoHS Compliant



V

Model Number Legend

V-[1][2]-[3][4][5][6]

1. Ratings
21: 21 A at 250 VAC
16: 16 A at 250 VAC
15: 15 A at 250 VAC
10: 10 A at 250 VAC

2. Actuator
None : Pin plunger
1 : Short hinge lever
2 : Hinge lever
3 : Long hinge lever
4 : Simulated roller lever
5 : Short hinge roller lever
6 : Hinge roller lever

3. Contact Form
1: SPDT
2: SPST-NC
3: SPST-NO

4. Terminals
A : Solder terminals
C2 : Quick-connect terminals (#187)
C : Quick-connect terminals (#250)

5. Maximum Operating Force
6: 3.92 N {400 gf}
5: 1.96 N {200 gf}
4: 0.98 N {100 gf}
Note: These values are for the pin plunger models.

6. Heat Resistance
None : Standard (80°C)
-T : Heat-resistive (150°C)

List of Models

Thermoplastic Case

Actuator	Terminals	Contact form	Ratings Maximum operating force (OF)	21A	16A
Pin plunger	Solder terminals (A)	SPDT	3.92N	---	**V-16-1A6**
		SPST-NC		---	**V-16-2A6**
		SPST-NO		---	**V-16-3A6**
		SPDT	1.96N	---	**V-16-1A5**
		SPST-NC		---	**V-16-2A5**
		SPST-NO		---	**V-16-3A5**
		SPDT	0.98N	---	---
		SPST-NC		---	---
		SPST-NO		---	---
	Quick-connect terminals (#187) (C2)	SPDT	3.92N	---	**V-16-1C26**
		SPST-NC		---	**V-16-2C26**
		SPST-NO		---	**V-16-3C26**
		SPDT	1.96N	---	**V-16-1C25**
		SPST-NC		---	**V-16-2C25**
		SPST-NO		---	**V-16-3C25**
		SPDT	0.98N	---	---
		SPST-NC		---	---
		SPST-NO		---	---
	Quick-connect terminals (#250) (C)	SPDT	3.92N	**V-21-1C6**	**V-16-1C6**
		SPST-NC		**V-21-2C6**	**V-16-2C6**
		SPST-NO		**V-21-3C6**	**V-16-3C6**
		SPDT	1.96N	---	**V-16-1C5**
		SPST-NC		---	**V-16-2C5**
		SPST-NO		---	**V-16-3C5**
		SPDT	0.98N	---	---
		SPST-NC		---	---
		SPST-NO		---	---

Refer to **"Micro Switch Common Accessories"** for Separators (sold separately), Actuators (sold separately) and Terminal Connectors (sold separately).

Actuator	Terminals	Contact form	Ratings: Maximum operating force (OF)	21A	16A
Short hinge lever 	Solder terminals (A)	SPDT	3.92N	---	**V-161-1A6**
		SPST-NC		---	**V-161-2A6**
		SPST-NO		---	**V-161-3A6**
		SPDT	1.96N	---	**V-161-1A5**
		SPST-NC		---	**V-161-2A5**
		SPST-NO		---	**V-161-3A5**
		SPDT	0.98N	---	---
		SPST-NC		---	---
		SPST-NO		---	---
	Quick-connect terminals (#187) (C2)	SPDT	3.92N	---	**V-161-1C26**
		SPST-NC		---	**V-161-2C26**
		SPST-NO		---	**V-161-3C26**
		SPDT	1.96N	---	**V-161-1C25**
		SPST-NC		---	**V-161-2C25**
		SPST-NO		---	**V-161-3C25**
		SPDT	0.98N	---	---
		SPST-NC		---	---
		SPST-NO		---	---
	Quick-connect terminals (#250) (C)	SPDT	3.92N	**V-211-1C6**	**V-161-1C6**
		SPST-NC		**V-211-2C6**	**V-161-2C6**
		SPST-NO		**V-211-3C6**	**V-161-3C6**
		SPDT	1.96N	---	**V-161-1C5**
		SPST-NC		---	**V-161-2C5**
		SPST-NO		---	**V-161-3C5**
		SPDT	0.98N	---	---
		SPST-NC		---	---
		SPST-NO		---	---
Hinge lever 	Solder terminals (A)	SPDT	2.45N	---	**V-162-1A6**
		SPST-NC		---	**V-162-2A6**
		SPST-NO		---	**V-162-3A6**
		SPDT	1.23N	---	**V-162-1A5**
		SPST-NC		---	**V-162-2A5**
		SPST-NO		---	**V-162-3A5**
		SPDT	0.59N	---	---
		SPST-NC		---	---
		SPST-NO		---	---
	Quick-connect terminals (#187) (C2)	SPDT	2.45N	---	**V-162-1C26**
		SPST-NC		---	**V-162-2C26**
		SPST-NO		---	**V-162-3C26**
		SPDT	1.23N	---	**V-162-1C25**
		SPST-NC		---	**V-162-2C25**
		SPST-NO		---	**V-162-3C25**
		SPDT	0.59N	---	---
		SPST-NC		---	---
		SPST-NO		---	---
	Quick-connect terminals (#250) (C)	SPDT	2.45N	**V-212-1C6**	**V-162-1C6**
		SPST-NC		**V-212-2C6**	**V-162-2C6**
		SPST-NO		**V-212-3C6**	**V-162-3C6**
		SPDT	1.23N	---	**V-162-1C5**
		SPST-NC		---	**V-162-2C5**
		SPST-NO		---	**V-162-3C5**
		SPDT	0.59N	---	---
		SPST-NC		---	---
		SPST-NO		---	---

Refer to **"Micro Switch Common Accessories"** for Separators (sold separately), Actuators (sold separately) and Terminal Connectors (sold separately).

Actuator	Terminals	Contact form	Ratings: Maximum operating force (OF)	21A	16A
Long hinge lever 	Solder terminals (A)	SPDT	1.27N	---	**V-163-1A6**
		SPST-NC		---	**V-163-2A6**
		SPST-NO		---	**V-163-3A6**
		SPDT	0.69N	---	**V-163-1A5**
		SPST-NC		---	**V-163-2A5**
		SPST-NO		---	**V-163-3A5**
		SPDT	0.34N	---	---
		SPST-NC		---	---
		SPST-NO		---	---
	Quick-connect terminals (#187) (C2)	SPDT	1.27N	---	**V-163-1C26**
		SPST-NC		---	**V-163-2C26**
		SPST-NO		---	**V-163-3C26**
		SPDT	0.69N	---	**V-163-1C25**
		SPST-NC		---	**V-163-2C25**
		SPST-NO		---	**V-163-3C25**
		SPDT	0.34N	---	---
		SPST-NC		---	---
		SPST-NO		---	---
	Quick-connect terminals (#250) (C)	SPDT	1.27N	**V-213-1C6**	**V-163-1C6**
		SPST-NC		**V-213-2C6**	**V-163-2C6**
		SPST-NO		**V-213-3C6**	**V-163-3C6**
		SPDT	0.69N	---	**V-163-1C5**
		SPST-NC		---	**V-163-2C5**
		SPST-NO		---	**V-163-3C5**
		SPDT	0.34N	---	---
		SPST-NC		---	---
		SPST-NO		---	---
Simulated roller lever 	Solder terminals (A)	SPDT	2.45N	---	**V-164-1A6**
		SPST-NC		---	**V-164-2A6**
		SPST-NO		---	**V-164-3A6**
		SPDT	1.23N	---	**V-164-1A5**
		SPST-NC		---	**V-164-2A5**
		SPST-NO		---	**V-164-3A5**
		SPDT	0.59N	---	---
		SPST-NC		---	---
		SPST-NO		---	---
	Quick-connect terminals (#187) (C2)	SPDT	2.45N	---	**V-164-1C26**
		SPST-NC		---	**V-164-2C26**
		SPST-NO		---	**V-164-3C26**
		SPDT	1.23N	---	**V-164-1C25**
		SPST-NC		---	**V-164-2C25**
		SPST-NO		---	**V-164-3C25**
		SPDT	0.59N	---	---
		SPST-NC		---	---
		SPST-NO		---	---
	Quick-connect terminals (#250) (C)	SPDT	2.45N	**V-214-1C6**	**V-164-1C6**
		SPST-NC		**V-214-2C6**	**V-164-2C6**
		SPST-NO		**V-214-3C6**	**V-164-3C6**
		SPDT	1.23N	---	**V-164-1C5**
		SPST-NC		---	**V-164-2C5**
		SPST-NO		---	**V-164-3C5**
		SPDT	0.59N	---	---
		SPST-NC		---	---
		SPST-NO		---	---

Refer to **"Micro Switch Common Accessories"** for Separators (sold separately), Actuators (sold separately) and Terminal Connectors (sold separately).

Actuator	Terminals	Contact form	Ratings Maximum operating force (OF)	21A	16A
Short hinge roller lever 	Solder terminals (A)	SPDT	4.71N	---	**V-165-1A6**
		SPST-NC		---	**V-165-2A6**
		SPST-NO		---	**V-165-3A6**
		SPDT	2.35N	---	**V-165-1A5**
		SPST-NC		---	**V-165-2A5**
		SPST-NO		---	**V-165-3A5**
		SPDT	1.18N	---	---
		SPST-NC		---	---
		SPST-NO		---	---
	Quick-connect terminals (#187) (C2)	SPDT	4.71N	---	**V-165-1C26**
		SPST-NC		---	**V-165-2C26**
		SPST-NO		---	**V-165-3C26**
		SPDT	2.35N	---	**V-165-1C25**
		SPST-NC		---	**V-165-2C25**
		SPST-NO		---	**V-165-3C25**
		SPDT	1.18N	---	---
		SPST-NC		---	---
		SPST-NO		---	---
	Quick-connect terminals (#250) (C)	SPDT	4.71N	**V-215-1C6**	**V-165-1C6**
		SPST-NC		**V-215-2C6**	**V-165-2C6**
		SPST-NO		**V-215-3C6**	**V-165-3C6**
		SPDT	2.35N	---	**V-165-1C5**
		SPST-NC		---	**V-165-2C5**
		SPST-NO		---	**V-165-3C5**
		SPDT	1.18N	---	---
		SPST-NC		---	---
		SPST-NO		---	---
Hinge roller lever 	Solder terminals (A)	SPDT	2.45N	---	**V-166-1A6**
		SPST-NC		---	**V-166-2A6**
		SPST-NO		---	**V-166-3A6**
		SPDT	1.23N	---	**V-166-1A5**
		SPST-NC		---	**V-166-2A5**
		SPST-NO		---	**V-166-3A5**
		SPDT	0.59N	---	---
		SPST-NC		---	---
		SPST-NO		---	---
	Quick-connect terminals (#187) (C2)	SPDT	2.45N	---	**V-166-1C26**
		SPST-NC		---	**V-166-2C26**
		SPST-NO		---	**V-166-3C26**
		SPDT	1.23N	---	**V-166-1C25**
		SPST-NC		---	**V-166-2C25**
		SPST-NO		---	**V-166-3C25**
		SPDT	0.59N	---	---
		SPST-NC		---	---
		SPST-NO		---	---
	Quick-connect terminals (#250) (C)	SPDT	2.45N	**V-216-1C6**	**V-166-1C6**
		SPST-NC		**V-216-2C6**	**V-166-2C6**
		SPST-NO		**V-216-3C6**	**V-166-3C6**
		SPDT	1.23N	---	**V-166-1C5**
		SPST-NC		---	**V-166-2C5**
		SPST-NO		---	**V-166-3C5**
		SPDT	0.59N	---	
		SPST-NC		---	
		SPST-NO		---	

Refer to **"Micro Switch Common Accessories"** for Separators (sold separately), Actuators (sold separately) and Terminal Connectors (sold separately).

V

Thermosetting case

Actuator	Terminals	Contact form	Ratings Maximum operating force (OF)	15A	10A	Heat-resistive 15A	Heat-resistive 10A
Pin plunger	Solder terminals (A)	SPDT	3.92N	V-15-1A6	---	V-15-1A6-T	---
		SPST-NC		V-15-2A6	---	---	---
		SPST-NO		V-15-3A6	---	---	---
		SPDT	1.96N	V-15-1A5	V-10-1A5	V-15-1A5-T	V-10-1A5-T
		SPST-NC		V-15-2A5	V-10-2A5	---	---
		SPST-NO		V-15-3A5	V-10-3A5	---	---
		SPDT	0.98N	---	V-10-1A4	---	V-10-1A4-T
		SPST-NC		---	V-10-2A4	---	V-10-2A4-T
		SPST-NO		---	V-10-3A4	---	V-10-3A4-T
	Quick-connect terminals (#187) (C2)	SPDT	3.92N	V-15-1C26	---	V-15-1C26-T	---
		SPST-NC		V-15-2C26	---	---	---
		SPST-NO		V-15-3C26	---	---	---
		SPDT	1.96N	V-15-1C25	V-10-1C25	V-15-1C25-T	V-10-1C25-T
		SPST-NC		V-15-2C25	V-10-2C25	---	---
		SPST-NO		V-15-3C25	V-10-3C25	---	---
		SPDT	0.98N	---	V-10-1C24	---	V-10-1C24-T
		SPST-NC		---	V-10-2C24	---	---
		SPST-NO		---	V-10-3C24	---	---
	Quick-connect terminals (#250) (C)	SPDT	3.92N	V-15-1C6	---	V-15-1C6-T	---
		SPST-NC		V-15-2C6	---	---	---
		SPST-NO		V-15-3C6	---	---	---
		SPDT	1.96N	V-15-1C5	V-10-1C5	V-15-1C5-T	V-10-1C5-T
		SPST-NC		V-15-2C5	V-10-2C5	---	---
		SPST-NO		V-15-3C5	V-10-3C5	---	---
		SPDT	0.98N	---	V-10-1C4	---	V-10-1C4-T
		SPST-NC		---	V-10-2C4	---	---
		SPST-NO		---	V-10-3C4	---	---
Short hinge lever	Solder terminals (A)	SPDT	3.92N	V-151-1A6	---	V-151-1A6-T	---
		SPST-NC		V-151-2A6	---	---	---
		SPST-NO		V-151-3A6	---	---	---
		SPDT	1.96N	V-151-1A5	V-101-1A5	V-151-1A5-T	V-101-1A5-T
		SPST-NC		V-151-2A5	V-101-2A5	---	---
		SPST-NO		V-151-3A5	V-101-3A5	---	---
		SPDT	0.98N	---	V-101-1A4	---	V-101-1A4-T
		SPST-NC		---	V-101-2A4	---	---
		SPST-NO		---	V-101-3A4	---	---
	Quick-connect terminals (#187) (C2)	SPDT	3.92N	V-151-1C26	---	V-151-1C26-T	---
		SPST-NC		V-151-2C26	---	---	---
		SPST-NO		V-151-3C26	---	---	---
		SPDT	1.96N	V-151-1C25	V-101-1C25	V-151-1C25-T	V-101-1C25-T
		SPST-NC		V-151-2C25	V-101-2C25	---	---
		SPST-NO		V-151-3C25	V-101-3C25	---	---
		SPDT	0.98N	---	V-101-1C24	---	V-101-1C24-T
		SPST-NC		---	V-101-2C24	---	---
		SPST-NO		---	V-101-3C24	---	---
	Quick-connect terminals (#250) (C)	SPDT	3.92N	V-151-1C6	---	V-151-1C6-T	---
		SPST-NC		V-151-2C6	---	---	---
		SPST-NO		V-151-3C6	---	---	---
		SPDT	1.96N	V-151-1C5	V-101-1C5	V-151-1C5-T	V-101-1C5-T
		SPST-NC		V-151-2C5	V-101-2C5	---	---
		SPST-NO		V-151-3C5	V-101-3C5	---	---
		SPDT	0.98N	---	V-101-1C4	---	V-101-1C4-T
		SPST-NC		---	V-101-2C4	---	---
		SPST-NO		---	V-101-3C4	---	---

Refer to **"Micro Switch Common Accessories"** for Separators (sold separately), Actuators (sold separately) and Terminal Connectors (sold separately).

Actuator	Terminals	Contact form	Ratings: Maximum operating force (OF)	15A	10A	Heat-resistive 15A	Heat-resistive 10A
Hinge lever	Solder terminals (A)	SPDT	2.45N	V-152-1A6	---	V-152-1A6-T	---
		SPST-NC		V-152-2A6	---	---	---
		SPST-NO		V-152-3A6	---	---	---
		SPDT	1.23N	V-152-1A5	V-102-1A5	V-152-1A5-T	V-102-1A5-T
		SPST-NC		V-152-2A5	V-102-2A5	---	---
		SPST-NO		V-152-3A5	V-102-3A5	---	---
		SPDT	0.59N	---	V-102-1A4	---	V-102-1A4-T
		SPST-NC		---	V-102-2A4	---	---
		SPST-NO		---	V-102-3A4	---	---
	Quick-connect terminals (#187) (C2)	SPDT	2.45N	V-152-1C26	---	V-152-1C26-T	---
		SPST-NC		V-152-2C26	---	---	---
		SPST-NO		V-152-3C26	---	---	---
		SPDT	1.23N	V-152-1C25	V-102-1C25	V-152-1C25-T	V-102-1C25-T
		SPST-NC		V-152-2C25	V-102-2C25	---	---
		SPST-NO		V-152-3C25	V-102-3C25	---	---
		SPDT	0.59N	---	V-102-1C24	---	V-102-1C24-T
		SPST-NC		---	V-102-2C24	---	---
		SPST-NO		---	V-102-3C24	---	---
	Quick-connect terminals (#250) (C)	SPDT	2.45N	V-152-1C6	---	V-152-1C6-T	---
		SPST-NC		V-152-2C6	---	---	---
		SPST-NO		V-152-3C6	---	---	---
		SPDT	1.23N	V-152-1C5	V-102-1C5	V-152-1C5-T	V-102-1C5-T
		SPST-NC		V-152-2C5	V-102-2C5	---	---
		SPST-NO		V-152-3C5	V-102-3C5	---	---
		SPDT	0.59N	---	V-102-1C4	---	V-102-1C4-T
		SPST-NC		---	V-102-2C4	---	---
		SPST-NO		---	V-102-3C4	---	---
Long hinge lever	Solder terminals (A)	SPDT	1.27N	V-153-1A6	---	V-153-1A6-T	---
		SPST-NC		V-153-2A6	---	---	---
		SPST-NO		V-153-3A6	---	---	---
		SPDT	0.69N	V-153-1A5	V-103-1A5	V-153-1A5-T	V-103-1A5-T
		SPST-NC		V-153-2A5	V-103-2A5	---	---
		SPST-NO		V-153-3A5	V-103-3A5	---	---
		SPDT	0.34N	---	V-103-1A4	---	V-103-1A4-T
		SPST-NC		---	V-103-2A4	---	---
		SPST-NO		---	V-103-3A4	---	---
	Quick-connect terminals (#187) (C2)	SPDT	1.27N	V-153-1C26	---	V-153-1C26-T	---
		SPST-NC		V-153-2C26	---	---	---
		SPST-NO		V-153-3C26	---	---	---
		SPDT	0.69N	V-153-1C25	V-103-1C25	V-153-1C25-T	V-103-1C25-T
		SPST-NC		V-153-2C25	V-103-2C25	---	---
		SPST-NO		V-153-3C25	V-103-3C25	---	---
		SPDT	0.34N	---	V-103-1C24	---	V-103-1C24-T
		SPST-NC		---	V-103-2C24	---	---
		SPST-NO		---	V-103-3C24	---	---
	Quick-connect terminals (#250) (C)	SPDT	1.27N	V-153-1C6	---	V-153-1C6-T	---
		SPST-NC		V-153-2C6	---	---	---
		SPST-NO		V-153-3C6	---	---	---
		SPDT	0.69N	V-153-1C5	V-103-1C5	V-153-1C5-T	V-103-1C5-T
		SPST-NC		V-153-2C5	V-103-2C5	---	---
		SPST-NO		V-153-3C5	V-103-3C5	---	---
		SPDT	0.34N	---	V-103-1C4	---	V-103-1C4-T
		SPST-NC		---	V-103-2C4	---	---
		SPST-NO		---	V-103-3C4	---	---

Refer to **"Micro Switch Common Accessories"** for Separators (sold separately), Actuators (sold separately) and Terminal Connectors (sold separately).

V

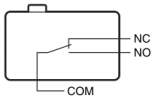
Actuator	Terminals	Contact form	Ratings Maximum operating force (OF)	15A	10A	Heat-resistive 15A	Heat-resistive 10A
Simulated roller lever 	Solder terminals (A)	SPDT	2.45N	V-154-1A6	---	V-154-1A6-T	---
		SPST-NC		V-154-2A6	---	---	---
		SPST-NO		V-154-3A6	---	---	---
		SPDT	1.23N	V-154-1A5	V-104-1A5	V-154-1A5-T	V-104-1A5-T
		SPST-NC		V-154-2A5	V-104-2A5	---	---
		SPST-NO		V-154-3A5	V-104-3A5	---	---
		SPDT	0.59N	---	V-104-1A4	---	V-104-1A4-T
		SPST-NC		---	V-104-2A4	---	---
		SPST-NO		---	V-104-3A4	---	---
	Quick-connect terminals (#187) (C2)	SPDT	2.45N	V-154-1C26	---	V-154-1C26-T	---
		SPST-NC		V-154-2C26	---	---	---
		SPST-NO		V-154-3C26	---	---	---
		SPDT	1.23N	V-154-1C25	V-104-1C25	V-154-1C25-T	V-104-1C25-T
		SPST-NC		V-154-2C25	V-104-2C25	---	---
		SPST-NO		V-154-3C25	V-104-3C25	---	---
		SPDT	0.59N	---	V-104-1C24	---	V-104-1C24-T
		SPST-NC		---	V-104-2C24	---	---
		SPST-NO		---	V-104-3C24	---	---
	Quick-connect terminals (#250) (C)	SPDT	2.45N	V-154-1C6	---	V-154-1C6-T	---
		SPST-NC		V-154-2C6	---	---	---
		SPST-NO		V-154-3C6	---	---	---
		SPDT	1.23N	V-154-1C5	V-104-1C5	V-154-1C5-T	V-104-1C5-T
		SPST-NC		V-154-2C5	V-104-2C5	---	---
		SPST-NO		V-154-3C5	V-104-3C5	---	---
		SPDT	0.59N	---	V-104-1C4	---	V-104-1C4-T
		SPST-NC		---	V-104-2C4	---	---
		SPST-NO		---	V-104-3C4	---	---
Short hinge roller lever 	Solder terminals (A)	SPDT	4.71N	V-155-1A6	---	V-155-1A6-T	---
		SPST-NC		V-155-2A6	---	---	---
		SPST-NO		V-155-3A6	---	---	---
		SPDT	2.35N	V-155-1A5	V-105-1A5	V-155-1A5-T	V-105-1A5-T
		SPST-NC		V-155-2A5	V-105-2A5	---	---
		SPST-NO		V-155-3A5	V-105-3A5	---	---
		SPDT	1.18N	---	V-105-1A4	---	V-105-1A4-T
		SPST-NC		---	V-105-2A4	---	---
		SPST-NO		---	V-105-3A4	---	---
	Quick-connect terminals (#187) (C2)	SPDT	4.71N	V-155-1C26	---	V-155-1C26-T	---
		SPST-NC		V-155-2C26	---	---	---
		SPST-NO		V-155-3C26	---	---	---
		SPDT	2.35N	V-155-1C25	V-105-1C25	V-155-1C25-T	V-105-1C25-T
		SPST-NC		V-155-2C25	V-105-2C25	---	---
		SPST-NO		V-155-3C25	V-105-3C25	---	---
		SPDT	1.18N	---	V-105-1C24	---	V-105-1C24-T
		SPST-NC		---	V-105-2C24	---	---
		SPST-NO		---	V-105-3C24	---	---
	Quick-connect terminals (#250) (C)	SPDT	4.71N	V-155-1C6	---	V-155-1C6-T	---
		SPST-NC		V-155-2C6	---	---	---
		SPST-NO		V-155-3C6	---	---	---
		SPDT	2.35N	V-155-1C5	V-105-1C5	V-155-1C5-T	V-105-1C5-T
		SPST-NC		V-155-2C5	V-105-2C5	---	---
		SPST-NO		V-155-3C5	V-105-3C5	---	---
		SPDT	1.18N	---	V-105-1C4	---	V-105-1C4-T
		SPST-NC		---	V-105-2C4	---	---
		SPST-NO		---	V-105-3C4	---	---

Refer to **"Micro Switch Common Accessories"** for Separators (sold separately), Actuators (sold separately) and Terminal Connectors (sold separately).

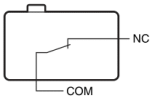
Actuator	Terminals	Contact form	Ratings Maximum operating force (OF)	15A	10A	Heat-resistive 15A	Heat-resistive 10A
Hinge roller lever 	Solder terminals (A)	SPDT	2.45N	**V-156-1A6**	---	**V-156-1A6-T**	---
		SPST-NC		**V-156-2A6**	---	---	---
		SPST-NO		**V-156-3A6**	---	---	---
		SPDT	1.23N	**V-156-1A5**	**V-106-1A5**	**V-156-1A5-T**	**V-106-1A5-T**
		SPST-NC		**V-156-2A5**	**V-106-2A5**	---	---
		SPST-NO		**V-156-3A5**	**V-106-3A5**	---	---
		SPDT	0.59N	---	**V-106-1A4**	---	**V-106-1A4-T**
		SPST-NC		---	**V-106-2A4**	---	---
		SPST-NO		---	**V-106-3A4**	---	---
	Quick-connect terminals (#187) (C2)	SPDT	2.45N	**V-156-1C26**	---	**V-156-1C26-T**	---
		SPST-NC		**V-156-2C26**	---	---	---
		SPST-NO		**V-156-3C26**	---	---	---
		SPDT	1.23N	**V-156-1C25**	**V-106-1C25**	**V-156-1C25-T**	**V-106-1C25-T**
		SPST-NC		**V-156-2C25**	**V-106-2C25**	---	---
		SPST-NO		**V-156-3C25**	**V-106-3C25**	---	---
		SPDT	0.59N	---	**V-106-1C24**	---	**V-106-1C24-T**
		SPST-NC		---	**V-106-2C24**	---	---
		SPST-NO		---	**V-106-3C24**	---	---
	Quick-connect terminals (#250) (C)	SPDT	2.45N	**V-156-1C6**	---	**V-156-1C6-T**	---
		SPST-NC		**V-156-2C6**	---	---	---
		SPST-NO		**V-156-3C6**	---	---	---
		SPDT	1.23N	**V-156-1C5**	**V-106-1C5**	**V-156-1C5-T**	**V-106-1C5-T**
		SPST-NC		**V-156-2C5**	**V-106-2C5**	---	---
		SPST-NO		**V-156-3C5**	**V-106-3C5**	---	---
		SPDT	0.59N	---	**V-106-1C4**	---	**V-106-1C4-T**
		SPST-NC		---	**V-106-2C4**	---	---
		SPST-NO		---	**V-106-3C4**	---	---

Contact form

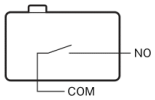
SPDT



SPST-NC



SPST-NO



Refer to **"Micro Switch Common Accessories"** for Separators (sold separately), Actuators (sold separately) and Terminal Connectors (sold separately).

Contact Specifications

Item	Model	V-21	V-16	V-15	V-10
Contact	Specification	Rivet			
	Material	Silver alloy			Silver
	Gap (standard value)	1 mm			
Inrush current	NC	50 A max.	40 A max.	30 A max.	24 A max.
	NO				
Minimum applicable load (reference value)		DC5V 160mA			

Ratings

Model	Item: Rated voltage	Resistive load
V-21	AC250V	21 A
	DC125V DC250V	0.6 A 0.3 A
V-16	AC250V	16 A
	DC125V DC250V	0.6 A 0.3 A
V-15	AC250V	15 A
	DC125V DC250V	0.6 A 0.3 A
V-10	AC250V	10 A
	DC125V DC250V	0.6 A 0.3 A

Note. The above rating values apply under the following test conditions.
(1) Ambient temperature: 20±2°C
(2) Ambient humidity: 65±5% RH
(3) Operating frequency: 30 operations/min

Approved Standards

UL (UL1054)/CSA (CSA C22.2 No.55)

Rated voltage / Model	V-21	V-16	V-15	V-10
125 VAC 250 VAC	21A 1/2HP	16A 1/2HP	15A 1/2HP	10A 1/2HP
125 VDC 250 VDC	0.6A 0.3A			

VDE (EN61058-1)

Consult your OMRON sales representative for specific models with VDE approvals.

Rated voltage / Model	V-21	V-16
AC250V	20(4)A	16(4)A

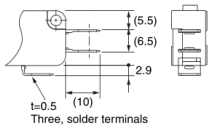
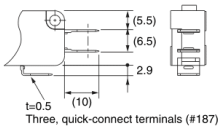
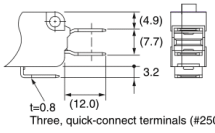
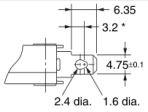
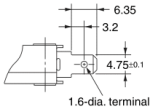
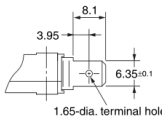
Testing conditions: 5E4 (50,000 operations), for models of V-21: T80 (0 to 80°C), for models of V-16: T105 (0 to 105°C)

Characteristics

Item	Model	V-10	V-15	V-16	V-21
Permissible operating speed		0.1mm to 1 m/s max. (pin plunger models)			
Permissible operating frequency	Mechanical	600 operations/min max. (pin plunger models)			
	Electrical	60 operations/min			
Insulation resistance		100MΩ min. (at 500 VDC with insulation tester)			
Contact resistance (initial value)		15mΩ max.			
Dielectric strength *1	Between terminals of the same polarity	AC1,000V 50/60Hz 1min			
	Between current-carrying metal parts and ground	AC1,500V 50/60Hz 1min	AC1,500V 50/60Hz 1min	AC2,000V 50/60Hz 1min	
	Between each terminals and non-current-carrying metal parts	AC1,500V 50/60Hz 1min	AC1,500V 50/60Hz 1min	AC2,000V 50/60Hz 1min	
Vibration resistance *2	Malfunction	10 to 55 Hz, 1.5-mm double amplitude			
Shock resistance *2	Durability	1,000 m/s² {approx. 100 G} max.			
	Malfunction	200 m/s² {approx. 20G} max.	300 m/s² {approx. 30 G} max.		
Durability *3	Mechanical	50,000,000 operations min. (60 operations/min)			
	Electrical	300,000 operations min. (30 operations/min) Heat resistive: 50,000 operations min (30 operations/min)	100,000 operations min. (30 operations/min) Heat resistive: 20,000 operations min (30 operations/min)	100,000 operations min. (30 operations/min)	
Degree of protection		IEC IP40			
Degree of protection against electric shock		Class I			
Proof tracking index (PTI)		175			
Ambient operating temperature		-25 to 105°C (Heat resistive: -25 to 150°C)		-25 to 105°C	-25 to 80°C
		(at ambient humidity of 60% max.) (with no icing or condensation)			
Ambient operating humidity		85% max. (for 5 to 35°C)			
Weight		Approx. 6.2g (pin plunger models)			

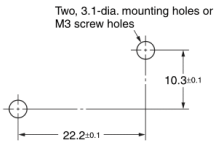
Note. The data given above are initial values.
*1. The dielectric strength shown in the table indicates a value for models with a Separator.
*2. For the pin plunger models, the above values apply for use at the free position and total travel position. For the lever models, they apply at the total travel position. Close or open circuit of the contact is shorter than 1 ms.
*3. For testing conditions, consult your OMRON sales representative.

Terminals and Apperance (Unit: mm)

Solder terminals (A)	Quick-connect terminals (#187) (C2)	Quick-connect terminals (#250) (C)
 Three, solder terminals	 Three, quick-connect terminals (#187)	 Three, quick-connect terminals (#250)
 * Indicates the length to the center of the 1.6-dia. holes	 1.6-dia. terminal hole	 1.65-dia. terminal hole

Note. The above is for the SPDT contact specifications. Two terminals will be available for SPST-NO or SPST-NC contact specifications. For terminal positions, refer to Contact form on page 9.

Mounting Holes (Unit: mm)



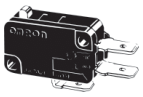
Dimensions and Operating Characteristics

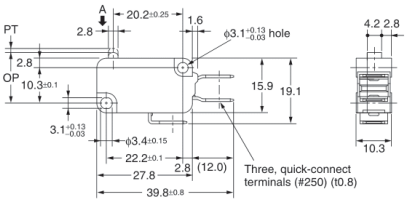
Thermoplastic Case V-21/-16 Models

The following illustrations and drawings are for quick-connect terminals #250 (terminals C). V models with a switching current of 16 A and 11 A incorporate solder terminals (A) and quick-connect terminals #187 (C2). These models are different from #250 models in terminal size only. Dimensions of solder terminals (A) and quick-connect terminal #187 (C2) are omitted. Please refer to the **"Terminals and Shapes"** on previous page.

The □ is replaced with the code for the terminals. See the "List of Models" for available combinations of shapes.

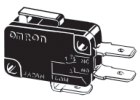
●Pin plunger
V-21-1□6
V-16-1□6
V-16-1□5

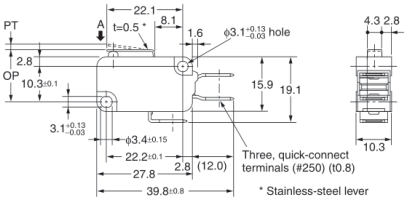




Operating characteristics	Model	V-21-1□6 V-16-1□6	V-16-1□5
OF max.		3.92N	1.96N
RF min.		0.78N	0.49N
PT max.		1.2mm	
OT min.		1.0mm	
MD max.		0.4mm	
OP		14.7±0.4mm	

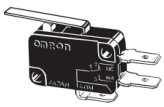
●Short hinge lever
V-211-1□6
V-161-1□6
V-161-1□5

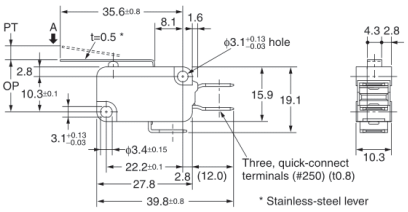




Operating characteristics	Model	V-211-1□6 V-161-1□6	V-161-1□5
OF max.		3.92N	1.96N
RF min.		0.49N	0.49N
PT max.		1.6mm	
OT min.		0.8mm	
MD max.		0.6mm	
OP		15.2±0.5mm	

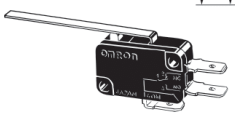
●Hinge lever
V-212-1□6
V-162-1□6
V-162-1□5

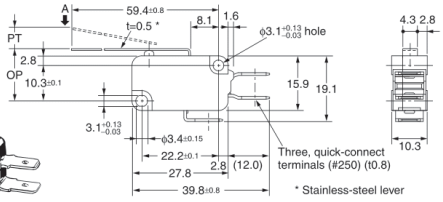




Operating characteristics	Model	V-212-1□6 V-162-1□6	V-162-1□5
OF max.		2.45N	1.23N
RF min.		0.25N	0.14N
PT max.		4.0mm	
OT min.		1.6mm	
MD max.		1.5mm	
OP		15.2±1.2mm	

●Long Hinge Lever Models
V-213-1□6
V-163-1□6
V-163-1□5



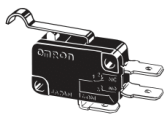


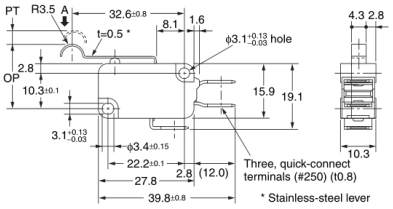
Operating characteristics	Model	V-213-1□6 V-163-1□6	V-163-1□5
OF max.		1.27N	0.69N
RF min.		0.12N	0.06N
PT max.		9.0mm	
OT min.		2.0mm	
MD max.		2.8mm	
OP		15.2 +2.6 −3.2 mm	

Note 1. Unless otherwise specified, a tolerance of ±0.4 mm applies to all dimensions.
Note 2. The operating characteristics are for operation in the A direction (⬇).

●Simulated roller lever

V-214-1□6
V-164-1□6
V-164-1□5

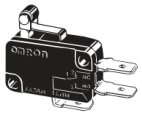


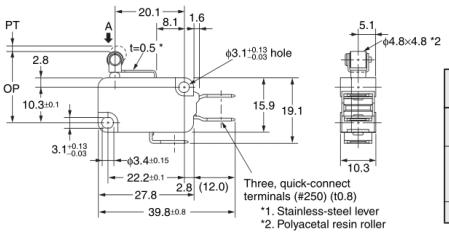


Operating characteristics	Model	V-214-1□6 V-164-1□6	V-164-1□5
OF max. RF min.		2.45N 0.25N	1.23N 0.14N
PT max. OT min. MD max.		4.0mm 1.6mm 1.5mm	
OP		18.7±1.2mm	

●Short hinge roller lever

V-215-1□6
V-165-1□6
V-165-1□5

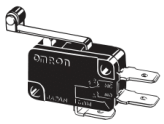


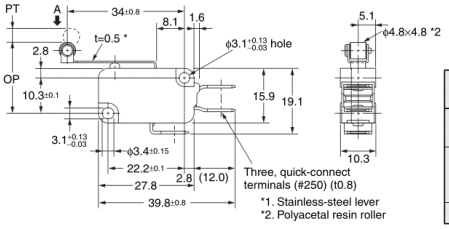


Operating characteristics	Model	V-215-1□6 V-165-1□6	V-165-1□5
OF max. RF min.		4.71N 0.49N	2.35N 0.49N
PT max. OT min. MD max.		1.6mm 0.8mm 0.6mm	
OP		20.7±0.6mm	

●Hinge roller lever

V-216-1□6
V-166-1□6
V-166-1□5





Operating characteristics	Model	V-216-1□6 V-166-1□6	V-166-1□5
OF max. RF min.		2.45N 0.25N	1.23N 0.14N
PT max. OT min. MD max.		4.0mm 1.6mm 1.5mm	
OP		20.7±1.2mm	

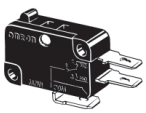
Note 1. Unless otherwise specified, a tolerance of ±0.4 mm applies to all dimensions.
Note 2. The operating characteristics are for operation in the A direction (⬇).

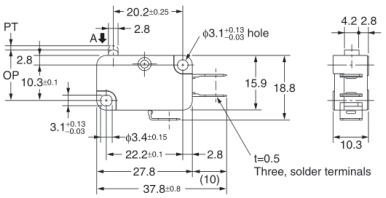
Thermosetting Case (V-15/V-10 Models) Applicable to both Standard (105°C) and Heat-resistive (150°C) models

The following dimensions and Operating Characteristics are for both "Not specified: Standard (105°C)" and "-T: Heat-resistive (150°C)" models.
The following illustrations and drawings are for solder terminals (Terminal A). V models with a switching current of 15A and 10A have quick-connect terminals #187 (C2). These models are different from solder terminal models in terminal size only. Illustrations for quick-connect terminals #187 (C2) are omitted. Please refer to **"Terminals and Shapes"** on page 8.

The □ is replaced with the code for the terminals.See the "List of Models" for available combinations of shapes.

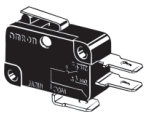
●Pin plunger
V-15-1□6
V-15-1□5
V-10-1□5
V-10-1□4

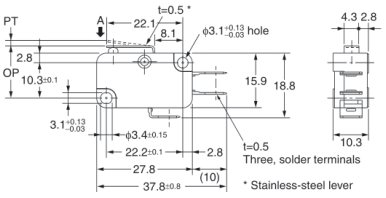




Operating characteristics / Model	V-15-1□6	V-15-1□5 V-10-1□5	V-10-1□4
OF max.	3.92N	1.96N	0.98N
RF min.	078N	0.49N	0.20N
PT max.	1.2mm		
OT min.	1.0mm		
MD max.	0.4mm		
OP	14.7±0.4mm		

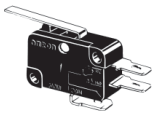
●Short hinge lever
V-151-1□6
V-151-1□5
V-101-1□5
V-101-1□4

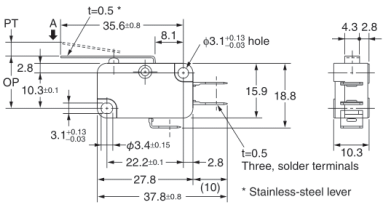




Operating characteristics / Model	V-151-1□6	V-151-1□5 V-101-1□5	V-101-1□4
OF max.	3.92N	1.96N	0.98N
RF min.	0.49N	0.49N	0.15N
PT max.	1.6mm		
OT min.	0.8mm		
MD max.	0.6mm		
OP	15.2±0.5mm		

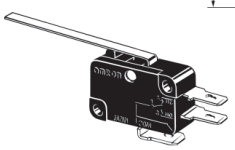
●Hinge lever
V-152-1□6
V-152-1□5
V-102-1□5
V-102-1□4

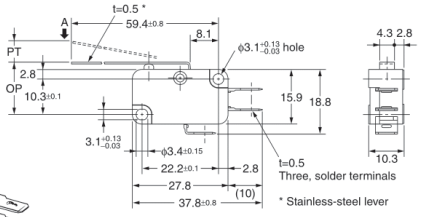




Operating characteristics / Model	V-152-1□6	V-152-1□5 V-102-1□5	V-102-1□4
OF max.	2.45N	1.23N	0.59N
RF min.	0.25N	0.14N	0.06N
PT max.	4.0mm		
OT min.	1.6mm		
MD max.	1.5mm		
OP	15.2±1.2mm		

●Long Hinge Lever Models
V-153-1□6
V-153-1□5
V-103-1□5
V-103-1□4

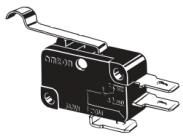


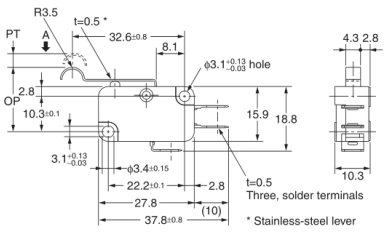


Operating characteristics / Model	V-153-1□6	V-153-1□5 V-103-1□5	V-103-1□4
OF max.	1.27N	0.69N	0.34N
RF min.	0.12N	0.06N	-
PT max.	9.0mm		9.0mm
OT min.	2.0mm		3.2mm
MD max.	2.8mm		2.8mm
OP	15.2 +2.6 −3.2 mm		15.2±2.6 mm

Note 1. Unless otherwise specified, a tolerance of ±0.4 mm applies to all dimensions.
Note 2. The operating characteristics are for operation in the A direction (⬇).

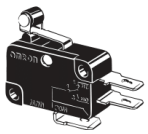
●Simulated roller lever
V-154-1□6
V-154-1□5
V-104-1□5
V-104-1□4

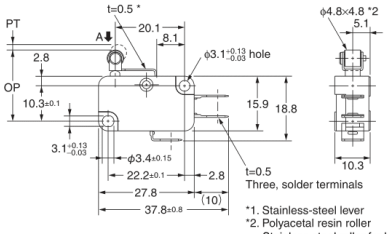




Operating characteristics	Model	V-154-1□6	V-154-1□5 V-104-1□5	V-104-1□4
OF max.		2.45N	1.23N	0.59N
RF min.		0.25N	0.14N	0.06N
PT max.		4.0mm		
OT min.		1.6mm		
MD max.		1.5mm		
OP		18.7±1.2mm		

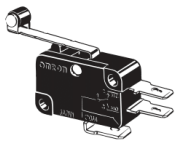
●Short hinge roller lever
V-155-1□6
V-155-1□5
V-105-1□5
V-105-1□4

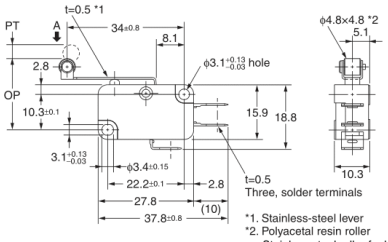




Operating characteristics	Model	V-155-1□6	V-155-1□5 V-105-1□5	V-105-1□4
OF max.		4.71N	2.35N	1.18N
RF min.		0.49N	0.49N	0.15N
PT max.		1.6mm		
OT min.		0.8mm		
MD max.		0.6mm		
OP		20.7±0.6mm		

●Hinge roller lever
V-156-1□6
V-156-1□5
V-106-1□5
V-106-1□4





Operating characteristics	Model	V-156-1□6	V-156-1□5 V-106-1□5	V-106-1□4
OF max.		2.45N	1.23N	0.59N
RF min.		0.25N	0.14N	0.06N
PT max.		4.0mm		
OT min.		1.6mm		
MD max.		1.5mm		
OP		20.7±1.2mm		

Note 1. Unless otherwise specified, a tolerance of ±0.4 mm applies to all dimensions.
Note 2. The operating characteristics are for operation in the A direction (⬇).

Precautions

★Please read "Common Precautions" for correct use.

Precautions for Safe Use

●Soldering

- Connecting to Solder Terminals
  Complete the soldering at the iron tip temperature of 250 to 350°C (60W) within 5 seconds, and do not apply any external force for 1 minute after soldering.
  Be sure to apply only the minimum required amount of flux.It may result in contact failure once the flux penetrates into the internal part of the Switch.
- Connecting to Quick-connect Terminals #187
  Insert the receptacle of quick-connect terminal #187 straight toward the terminal.
  Applying excessive external force horizontally or vertically may cause deformation of terminals and may damage the housings.
- Connecting to Quick-connect Terminals #250
  Insert the receptacle of quick-connect terminal #250 straight toward the terminal.
  Applying excessive external force horizontally or vertically may cause deformation of terminals and may damage the housings.

Precautions for Correct Use

●Mounting

Use M3 mounting screw with plane washers or spring washers to securely mount the Switch.Tighten the screws to a torque of 0.39 to 0.59N·m {4 to 6 kgf·cm}.

V

OMRON Corporation
Electronic and Mechanical Components Company **Contact: www.omron.com/ecb**

Cat. No. B010-E1-14
0615(0207)(O)

Appendix J – PCA9685 Datasheet

PCA9685

16-channel, 12-bit PWM Fm+ I²C-bus LED controller

Rev. 4 — 16 April 2015 Product data sheet

1. General description

The PCA9685 is an I²C-bus controlled 16-channel LED controller optimized for Red/Green/Blue/Amber (RGBA) color backlighting applications. Each LED output has its own 12-bit resolution (4096 steps) fixed frequency individual PWM controller that operates at a programmable frequency from a typical of 24 Hz to 1526 Hz with a duty cycle that is adjustable from 0 % to 100 % to allow the LED to be set to a specific brightness value. All outputs are set to the same PWM frequency.

Each LED output can be off or on (no PWM control), or set at its individual PWM controller value. The LED output driver is programmed to be either open-drain with a 25 mA current sink capability at 5 V or totem pole with a 25 mA sink, 10 mA source capability at 5 V. The PCA9685 operates with a supply voltage range of 2.3 V to 5.5 V and the inputs and outputs are 5.5 V tolerant. LEDs can be directly connected to the LED output (up to 25 mA, 5.5 V) or controlled with external drivers and a minimum amount of discrete components for larger current or higher voltage LEDs.

The PCA9685 is in the new Fast-mode Plus (Fm+) family. Fm+ devices offer higher frequency (up to 1 MHz) and more densely populated bus operation (up to 4000 pF).

Although the PCA9635 and PCA9685 have many similar features, the PCA9685 has some unique features that make it more suitable for applications such as LCD or LED backlighting and Ambilight:

- The PCA9685 allows staggered LED output on and off times to minimize current surges. The on and off time delay is independently programmable for each of the 16 channels. This feature is not available in PCA9635.
- The PCA9685 has 4096 steps (12-bit PWM) of individual LED brightness control. The PCA9635 has only 256 steps (8-bit PWM).
- When multiple LED controllers are incorporated in a system, the PWM pulse widths between multiple devices may differ if PCA9635s are used. The PCA9685 has a programmable prescaler to adjust the PWM pulse widths of multiple devices.
- The PCA9685 has an external clock input pin that will accept user-supplied clock (50 MHz max.) in place of the internal 25 MHz oscillator. This feature allows synchronization of multiple devices. The PCA9635 does not have external clock input feature.
- Like the PCA9635, PCA9685 also has a built-in oscillator for the PWM control. However, the frequency used for PWM control in the PCA9685 is adjustable from about 24 Hz to 1526 Hz as compared to the typical 97.6 kHz frequency of the PCA9635. This allows the use of PCA9685 with external power supply controllers. All bits are set at the same frequency.
- The Power-On Reset (POR) default state of LEDn output pins is LOW in the case of PCA9685. It is HIGH for PCA9635.

The active LOW Output Enable input pin ($\overline{OE}$) allows asynchronous control of the LED outputs and can be used to set all the outputs to a defined I²C-bus programmable logic state. The $\overline{OE}$ can also be used to externally 'pulse width modulate' the outputs, which is useful when multiple devices need to be dimmed or blinked together using software control.

Software programmable LED All Call and three Sub Call I²C-bus addresses allow all or defined groups of PCA9685 devices to respond to a common I²C-bus address, allowing for example, all red LEDs to be turned on or off at the same time or marquee chasing effect, thus minimizing I²C-bus commands. Six hardware address pins allow up to 62 devices on the same bus.

The Software Reset (SWRST) General Call allows the master to perform a reset of the PCA9685 through the I²C-bus, identical to the Power-On Reset (POR) that initializes the registers to their default state causing the outputs to be set LOW. This allows an easy and quick way to reconfigure all device registers to the same condition via software.

2. Features and benefits

- 16 LED drivers. Each output programmable at:
  - Off
  - On
  - Programmable LED brightness
  - Programmable LED turn-on time to help reduce EMI
- 1 MHz Fast-mode Plus compatible I²C-bus interface with 30 mA high drive capability on SDA output for driving high capacitive buses
- 4096-step (12-bit) linear programmable brightness per LED output varying from fully off (default) to maximum brightness
- LED output frequency (all LEDs) typically varies from 24 Hz to 1526 Hz (Default of 1Eh in PRE_SCALE register results in a 200 Hz refresh rate with oscillator clock of 25 MHz.)
- Sixteen totem pole outputs (sink 25 mA and source 10 mA at 5 V) with software programmable open-drain LED outputs selection (default at totem pole). No input function.
- Output state change programmable on the Acknowledge or the STOP Command to update outputs byte-by-byte or all at the same time (default to 'Change on STOP').
- Active LOW Output Enable ($\overline{OE}$) input pin. LEDn outputs programmable to logic 1, logic 0 (default at power-up) or 'high-impedance' when $\overline{OE}$ is HIGH.
- 6 hardware address pins allow 62 PCA9685 devices to be connected to the same I²C-bus
- Toggling $\overline{OE}$ allows for hardware LED blinking
- 4 software programmable I²C-bus addresses (one LED All Call address and three LED Sub Call addresses) allow groups of devices to be addressed at the same time in any combination (for example, one register used for 'All Call' so that all the PCA9685s on the I²C-bus can be addressed at the same time and the second register used for three different addresses so that 1⁄3 of all devices on the bus can be addressed at the same time in a group). Software enable and disable for these I²C-bus address.
- Software Reset feature (SWRST General Call) allows the device to be reset through the I²C-bus

- 25 MHz typical internal oscillator requires no external components
- External 50 MHz (max.) clock input
- Internal power-on reset
- Noise filter on SDA/SCL inputs
- Edge rate control on outputs
- No output glitches on power-up
- Supports hot insertion
- Low standby current
- Operating power supply voltage range of 2.3 V to 5.5 V
- 5.5 V tolerant inputs
- −40 °C to +85 °C operation
- ESD protection exceeds 2000 V HBM per JESD22-A114, 200 V MM per JESD22-A115 and 1000 V CDM per JESD22-C101
- Latch-up testing is done to JEDEC Standard JESD78 which exceeds 100 mA
- Packages offered: TSSOP28, HVQFN28

3. Applications

- RGB or RGBA LED drivers
- LED status information
- LED displays
- LCD backlights
- Keypad backlights for cellular phones or handheld devices

4. Ordering information

Table 1. Ordering information

Type number	Topside mark	Package		
		Name	Description	Version
PCA9685PW	PCA9685PW	TSSOP28	plastic thin shrink small outline package; 28 leads; body width 4.4 mm	SOT361-1
PCA9685PW/Q900[1]	PCA9685PW	TSSOP28	plastic thin shrink small outline package; 28 leads; body width 4.4 mm	SOT361-1
PCA9685BS	P9685	HVQFN28	plastic thermal enhanced very thin quad flat package; no leads; 28 terminals; body 6 × 6 × 0.85 mm	SOT788-1

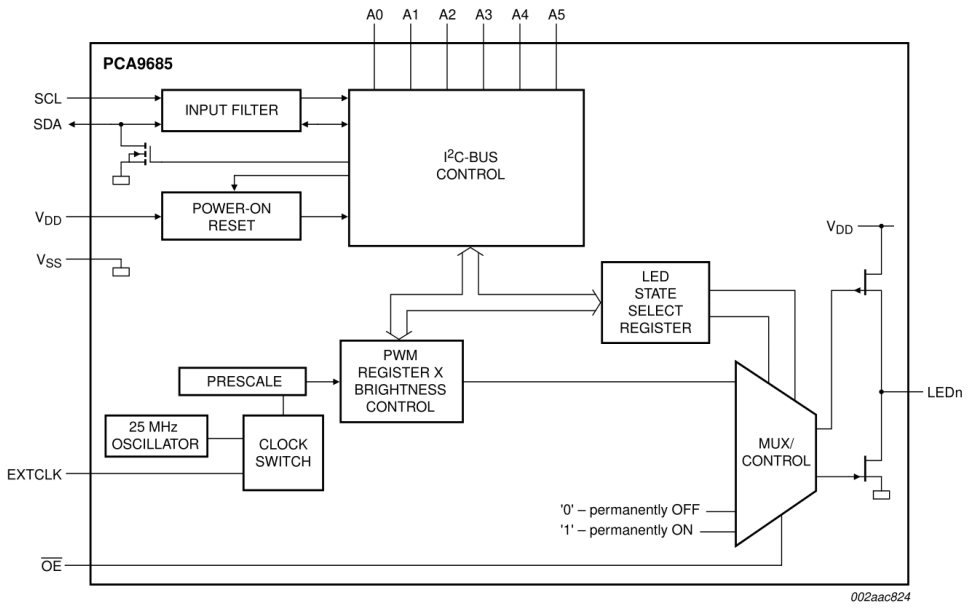
[1] PCA9685PW/Q900 is AEC-Q100 compliant. Contact **i2c.support@nxp.com** for PPAP.

4.1 Ordering options

Table 2. Ordering options

Type number	Orderable part number	Package	Packing method	Minimum order quantity	Temperature
PCA9685PW	PCA9685PW,118	TSSOP28	REEL 13" Q1/T1 *STANDARD MARK SMD	2500	T_{amb} = −40 °C to +85 °C
PCA9685PW/Q900	PCA9685PW/Q900,118	TSSOP28	REEL 13" Q1/T1 *STANDARD MARK SMD	2500	T_{amb} = −40 °C to +85 °C
PCA9685BS	PCA9685BS,118	HVQFN28	REEL 13" Q1/T1 *STANDARD MARK SMD	4000	T_{amb} = −40 °C to +85 °C

5. Block diagram



Remark: Only one LED output shown for clarity.

Fig 1. Block diagram of PCA9685

6. Pinning information

6.1 Pinning

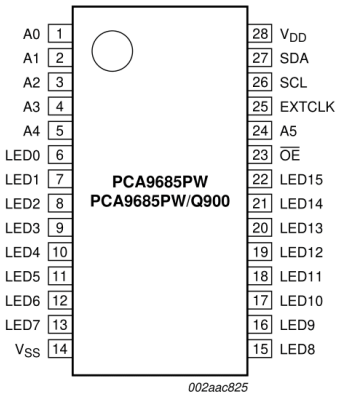


Fig 2. Pin configuration for TSSOP28

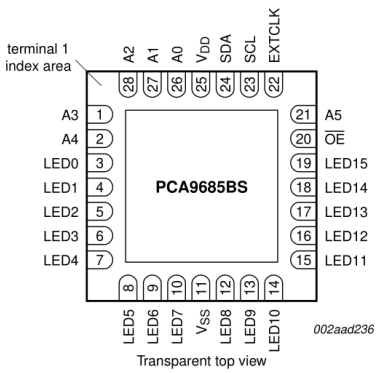


Fig 3. Pin configuration for HVQFN28

6.2 Pin description

Table 3. Pin description

Symbol	Pin		Type	Description
	TSSOP28	HVQFN28		
A0	1	26	I	address input 0
A1	2	27	I	address input 1
A2	3	28	I	address input 2
A3	4	1	I	address input 3
A4	5	2	I	address input 4
LED0	6	3	O	LED driver 0
LED1	7	4	O	LED driver 1
LED2	8	5	O	LED driver 2
LED3	9	6	O	LED driver 3
LED4	10	7	O	LED driver 4
LED5	11	8	O	LED driver 5
LED6	12	9	O	LED driver 6
LED7	13	10	O	LED driver 7
V_{SS}	14	11[1]	power supply	supply ground
LED8	15	12	O	LED driver 8
LED9	16	13	O	LED driver 9
LED10	17	14	O	LED driver 10
LED11	18	15	O	LED driver 11

Table 3. Pin description …continued

Symbol	Pin		Type	Description
	TSSOP28	HVQFN28		
LED12	19	16	O	LED driver 12
LED13	20	17	O	LED driver 13
LED14	21	18	O	LED driver 14
LED15	22	19	O	LED driver 15
$\overline{OE}$	23	20	I	active LOW output enable
A5	24	21	I	address input 5
EXTCLK	25	22	I	external clock input[2]
SCL	26	23	I	serial clock line
SDA	27	24	I/O	serial data line
V_{DD}	28	25	power supply	supply voltage

[1] HVQFN28 package die supply ground is connected to both V_{SS} pin and exposed center pad. V_{SS} pin must be connected to supply ground for proper device operation. For enhanced thermal, electrical, and board level performance, the exposed pad needs to be soldered to the board using a corresponding thermal pad on the board and for proper heat conduction through the board, thermal vias need to be incorporated in the PCB in the thermal pad region.

[2] This pin must be grounded when this feature is not used.

7. Functional description

Refer to Figure 1 “Block diagram of PCA9685”.

7.1 Device addresses

Following a START condition, the bus master must output the address of the slave it is accessing.

There are a maximum of 64 possible programmable addresses using the 6 hardware address pins. Two of these addresses, Software Reset and LED All Call, cannot be used because their default power-up state is ON, leaving a maximum of 62 addresses. Using other reserved addresses, as well as any other subcall address, will reduce the total number of possible addresses even further.

7.1.1 Regular I²C-bus slave address

The I²C-bus slave address of the PCA9685 is shown in Figure 4. To conserve power, no internal pull-up resistors are incorporated on the hardware selectable address pins and they must be pulled HIGH or LOW.

Remark: Using reserved I²C-bus addresses will interfere with other devices, but only if the devices are on the bus and/or the bus will be open to other I²C-bus systems at some later date. In a closed system where the designer controls the address assignment these addresses can be used since the PCA9685 treats them like any other address. The LED All Call, Software Reset and PCA9564 or PCA9665 slave address (if on the bus) can never be used for individual device addresses.

- PCA9685 LED All Call address (1110 000) and Software Reset (0000 0110) which are active on start-up

- PCA9564 (0000 000) or PCA9665 (1110 000) slave address which is active on start-up
- 'reserved for future use' I²C-bus addresses (0000 011, 1111 1XX)
- slave devices that use the 10-bit addressing scheme (1111 0XX)
- slave devices that are designed to respond to the General Call address (0000 000) which is used as the software reset address
- High-speed mode (Hs-mode) master code (0000 1XX)

Fig 4. Slave address

The last bit of the address byte defines the operation to be performed. When set to logic 1 a read is selected, while a logic 0 selects a write operation.

7.1.2 LED All Call I²C-bus address

- Default power-up value (ALLCALLADR register): E0h or 1110 000X
- Programmable through I²C-bus (volatile programming)
- At power-up, LED All Call I²C-bus address is enabled. PCA9685 sends an ACK when E0h (R/$\overline{W}$ = 0) or E1h (R/$\overline{W}$ = 1) is sent by the master.

See Section 7.3.7 "ALLCALLADR, LED All Call I²C-bus address" for more detail.

Remark: The default LED All Call I²C-bus address (E0h or 1110 000X) must not be used as a regular I²C-bus slave address since this address is enabled at power-up. All the PCA9685s on the I²C-bus will acknowledge the address if sent by the I²C-bus master.

7.1.3 LED Sub Call I²C-bus addresses

- 3 different I²C-bus addresses can be used
- Default power-up values:
  - SUBADR1 register: E2h or 1110 001X
  - SUBADR2 register: E4h or 1110 010X
  - SUBADR3 register: E8h or 1110 100X
- Programmable through I²C-bus (volatile programming)
- At power-up, Sub Call I²C-bus addresses are disabled. PCA9685 does not send an ACK when E2h (R/$\overline{W}$ = 0) or E3h (R/$\overline{W}$ = 1), E4h (R/$\overline{W}$ = 0) or E5h (R/$\overline{W}$ = 1), or E8h (R/$\overline{W}$ = 0) or E9h (R/$\overline{W}$ = 1) is sent by the master.

See Section 7.3.6 "SUBADR1 to SUBADR3, I²C-bus subaddress 1 to 3" for more detail.

Remark: The default LED Sub Call I²C-bus addresses may be used as regular I²C-bus slave addresses as long as they are disabled.

7.1.4 Software Reset I²C-bus address

The address shown in Figure 5 is used when a reset of the PCA9685 needs to be performed by the master. The Software Reset address (SWRST Call) must be used with R/$\overline{W}$ = logic 0. If R/$\overline{W}$ = logic 1, the PCA9685 does not acknowledge the SWRST. See Section 7.6 “Software reset” for more detail.

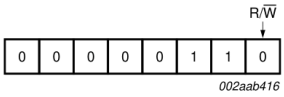


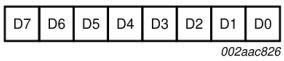
Fig 5. Software Reset address

Remark: The Software Reset I²C-bus address is a reserved address and cannot be used as a regular I²C-bus slave address or as an LED All Call or LED Sub Call address.

7.2 Control register

Following the successful acknowledgement of the slave address, LED All Call address or LED Sub Call address, the bus master will send a byte to the PCA9685, which will be stored in the Control register.

This register is used as a pointer to determine which register will be accessed.



reset state = 00h

Remark: The Control register does not apply to the Software Reset I²C-bus address.

Fig 6. Control register

7.3 Register definitions

Table 4. Register summary

Register# (decimal)	Register# (hex)	D7	D6	D5	D4	D3	D2	D1	D0	Name	Type	Function
0	00	0	0	0	0	0	0	0	0	MODE1	read/write	Mode register 1
1	01	0	0	0	0	0	0	0	1	MODE2	read/write	Mode register 2
2	02	0	0	0	0	0	0	1	0	SUBADR1	read/write	I²C-bus subaddress 1
3	03	0	0	0	0	0	0	1	1	SUBADR2	read/write	I²C-bus subaddress 2
4	04	0	0	0	0	0	1	0	0	SUBADR3	read/write	I²C-bus subaddress 3
5	05	0	0	0	0	0	1	0	1	ALLCALLADR	read/write	LED All Call I²C-bus address
6	06	0	0	0	0	0	1	1	0	LED0_ON_L	read/write	LED0 output and brightness control byte 0
7	07	0	0	0	0	0	1	1	1	LED0_ON_H	read/write	LED0 output and brightness control byte 1
8	08	0	0	0	0	1	0	0	0	LED0_OFF_L	read/write	LED0 output and brightness control byte 2
9	09	0	0	0	0	1	0	0	1	LED0_OFF_H	read/write	LED0 output and brightness control byte 3
10	0A	0	0	0	0	1	0	1	0	LED1_ON_L	read/write	LED1 output and brightness control byte 0
11	0B	0	0	0	0	1	0	1	1	LED1_ON_H	read/write	LED1 output and brightness control byte 1
12	0C	0	0	0	0	1	1	0	0	LED1_OFF_L	read/write	LED1 output and brightness control byte 2
13	0D	0	0	0	0	1	1	0	1	LED1_OFF_H	read/write	LED1 output and brightness control byte 3
14	0E	0	0	0	0	1	1	1	0	LED2_ON_L	read/write	LED2 output and brightness control byte 0
15	0F	0	0	0	0	1	1	1	1	LED2_ON_H	read/write	LED2 output and brightness control byte 1
16	10	0	0	0	1	0	0	0	0	LED2_OFF_L	read/write	LED2 output and brightness control byte 2
17	11	0	0	0	1	0	0	0	1	LED2_OFF_H	read/write	LED2 output and brightness control byte 3
18	12	0	0	0	1	0	0	1	0	LED3_ON_L	read/write	LED3 output and brightness control byte 0
19	13	0	0	0	1	0	0	1	1	LED3_ON_H	read/write	LED3 output and brightness control byte 1
20	14	0	0	0	1	0	1	0	0	LED3_OFF_L	read/write	LED3 output and brightness control byte 2
21	15	0	0	0	1	0	1	0	1	LED3_OFF_H	read/write	LED3 output and brightness control byte 3

Table 4. Register summary *…continued*

Register# (decimal)	Register# (hex)	D7	D6	D5	D4	D3	D2	D1	D0	Name	Type	Function
22	16	0	0	0	1	0	1	1	0	LED4_ON_L	read/write	LED4 output and brightness control byte 0
23	17	0	0	0	1	0	1	1	1	LED4_ON_H	read/write	LED4 output and brightness control byte 1
24	18	0	0	0	1	1	0	0	0	LED4_OFF_L	read/write	LED4 output and brightness control byte 2
25	19	0	0	0	1	1	0	0	1	LED4_OFF_H	read/write	LED4 output and brightness control byte 3
26	1A	0	0	0	1	1	0	1	0	LED5_ON_L	read/write	LED5 output and brightness control byte 0
27	1B	0	0	0	1	1	0	1	1	LED5_ON_H	read/write	LED5 output and brightness control byte 1
28	1C	0	0	0	1	1	1	0	0	LED5_OFF_L	read/write	LED5 output and brightness control byte 2
29	1D	0	0	0	1	1	1	0	1	LED5_OFF_H	read/write	LED5 output and brightness control byte 3
30	1E	0	0	0	1	1	1	1	0	LED6_ON_L	read/write	LED6 output and brightness control byte 0
31	1F	0	0	0	1	1	1	1	1	LED6_ON_H	read/write	LED6 output and brightness control byte 1
32	20	0	0	1	0	0	0	0	0	LED6_OFF_L	read/write	LED6 output and brightness control byte 2
33	21	0	0	1	0	0	0	0	1	LED6_OFF_H	read/write	LED6 output and brightness control byte 3
34	22	0	0	1	0	0	0	1	0	LED7_ON_L	read/write	LED7 output and brightness control byte 0
35	23	0	0	1	0	0	0	1	1	LED7_ON_H	read/write	LED7 output and brightness control byte 1
36	24	0	0	1	0	0	1	0	0	LED7_OFF_L	read/write	LED7 output and brightness control byte 2
37	25	0	0	1	0	0	1	0	1	LED7_OFF_H	read/write	LED7 output and brightness control byte 3
38	26	0	0	1	0	0	1	1	0	LED8_ON_L	read/write	LED8 output and brightness control byte 0
39	27	0	0	1	0	0	1	1	1	LED8_ON_H	read/write	LED8 output and brightness control byte 1
40	28	0	0	1	0	1	0	0	0	LED8_OFF_L	read/write	LED8 output and brightness control byte 2
41	29	0	0	1	0	1	0	0	1	LED8_OFF_H	read/write	LED8 output and brightness control byte 3

Table 4. Register summary *…continued*

Register# (decimal)	Register# (hex)	D7	D6	D5	D4	D3	D2	D1	D0	Name	Type	Function
42	2A	0	0	1	0	1	0	1	0	LED9_ON_L	read/write	LED9 output and brightness control byte 0
43	2B	0	0	1	0	1	0	1	1	LED9_ON_H	read/write	LED9 output and brightness control byte 1
44	2C	0	0	1	0	1	1	0	0	LED9_OFF_L	read/write	LED9 output and brightness control byte 2
45	2D	0	0	1	0	1	1	0	1	LED9_OFF_H	read/write	LED9 output and brightness control byte 3
46	2E	0	0	1	0	1	1	1	0	LED10_ON_L	read/write	LED10 output and brightness control byte 0
47	2F	0	0	1	0	1	1	1	1	LED10_ON_H	read/write	LED10 output and brightness control byte 1
48	30	0	0	1	1	0	0	0	0	LED10_OFF_L	read/write	LED10 output and brightness control byte 2
49	31	0	0	1	1	0	0	0	1	LED10_OFF_H	read/write	LED10 output and brightness control byte 3
50	32	0	0	1	1	0	0	1	0	LED11_ON_L	read/write	LED11 output and brightness control byte 0
51	33	0	0	1	1	0	0	1	1	LED11_ON_H	read/write	LED11 output and brightness control byte 1
52	34	0	0	1	1	0	1	0	0	LED11_OFF_L	read/write	LED11 output and brightness control byte 2
53	35	0	0	1	1	0	1	0	1	LED11_OFF_H	read/write	LED11 output and brightness control byte 3
54	36	0	0	1	1	0	1	1	0	LED12_ON_L	read/write	LED12 output and brightness control byte 0
55	37	0	0	1	1	0	1	1	1	LED12_ON_H	read/write	LED12 output and brightness control byte 1
56	38	0	0	1	1	1	0	0	0	LED12_OFF_L	read/write	LED12 output and brightness control byte 2
57	39	0	0	1	1	1	0	0	1	LED12_OFF_H	read/write	LED12 output and brightness control byte 3
58	3A	0	0	1	1	1	0	1	0	LED13_ON_L	read/write	LED13 output and brightness control byte 0
59	3B	0	0	1	1	1	0	1	1	LED13_ON_H	read/write	LED13 output and brightness control byte 1
60	3C	0	0	1	1	1	1	0	0	LED13_OFF_L	read/write	LED13 output and brightness control byte 2
61	3D	0	0	1	1	1	1	0	1	LED13_OFF_H	read/write	LED13 output and brightness control byte 3

Table 4. Register summary *...continued*

Register# (decimal)	Register# (hex)	D7	D6	D5	D4	D3	D2	D1	D0	Name	Type	Function
62	3E	0	0	1	1	1	1	1	0	LED14_ON_L	read/write	LED14 output and brightness control byte 0
63	3F	0	0	1	1	1	1	1	1	LED14_ON_H	read/write	LED14 output and brightness control byte 1
64	40	0	1	0	0	0	0	0	0	LED14_OFF_L	read/write	LED14 output and brightness control byte 2
65	41	0	1	0	0	0	0	0	1	LED14_OFF_H	read/write	LED14 output and brightness control byte 3
66	42	0	1	0	0	0	0	1	0	LED15_ON_L	read/write	LED15 output and brightness control byte 0
67	43	0	1	0	0	0	0	1	1	LED15_ON_H	read/write	LED15 output and brightness control byte 1
68	44	0	1	0	0	0	1	0	0	LED15_OFF_L	read/write	LED15 output and brightness control byte 2
69	45	0	1	0	0	0	1	0	1	LED15_OFF_H	read/write	LED15 output and brightness control byte 3
...	reserved for future use											
250	FA	1	1	1	1	1	0	1	0	ALL_LED_ON_L	write/read zero	load all the LEDn_ON registers, byte 0
251	FB	1	1	1	1	1	0	1	1	ALL_LED_ON_H	write/read zero	load all the LEDn_ON registers, byte 1
252	FC	1	1	1	1	1	1	0	0	ALL_LED_OFF_L	write/read zero	load all the LEDn_OFF registers, byte 0
253	FD	1	1	1	1	1	1	0	1	ALL_LED_OFF_H	write/read zero	load all the LEDn_OFF registers, byte 1
254	FE	1	1	1	1	1	1	1	0	PRE_SCALE[1]	read/write	prescaler for PWM output frequency
255	FF	1	1	1	1	1	1	1	1	TestMode[2]	read/write	defines the test mode to be entered
...	All further addresses are reserved for future use; reserved addresses will not be acknowledged.											

[1] Writes to PRE_SCALE register are blocked when SLEEP bit is logic 0 (MODE 1).

[2] Reserved. Writes to this register may cause unpredictable results.

Remark: Auto Increment past register 69 will point to MODE1 register (register 0). Auto Increment also works from register 250 to register 254, then rolls over to register 0.

7.3.1 Mode register 1, MODE1

Table 5. MODE1 - Mode register 1 (address 00h) bit description

*Legend: * default value.*

Bit	Symbol	Access	Value	Description
7	RESTART	R		Shows state of RESTART logic. See Section 7.3.1.1 for detail.
		W		User writes logic 1 to this bit to clear it to logic 0. A user write of logic 0 will have no effect. See Section 7.3.1.1 for detail.
			0*	Restart disabled.
			1	Restart enabled.
6	EXTCLK	R/W		To use the EXTCLK pin, this bit must be set by the following sequence: 1. Set the SLEEP bit in MODE1. This turns off the internal oscillator. 2. Write logic 1s to both the SLEEP and EXTCLK bits in MODE1. The switch is now made. The external clock can be active during the switch because the SLEEP bit is set. This bit is a 'sticky bit', that is, it cannot be cleared by writing a logic 0 to it. The EXTCLK bit can **only** be cleared by a power cycle or software reset. EXTCLK range is DC to 50 MHz. $refresh\_rate = \frac{EXTCLK}{4096 \times (prescale + 1)}$
			0*	Use internal clock.
			1	Use EXTCLK pin clock.
5	AI	R/W	0*	Register Auto-Increment disabled[1].
			1	Register Auto-Increment enabled.
4	SLEEP	R/W	0	Normal mode[2].
			1*	Low power mode. Oscillator off[3][4].
3	SUB1	R/W	0*	PCA9685 does not respond to I²C-bus subaddress 1.
			1	PCA9685 responds to I²C-bus subaddress 1.
2	SUB2	R/W	0*	PCA9685 does not respond to I²C-bus subaddress 2.
			1	PCA9685 responds to I²C-bus subaddress 2.
1	SUB3	R/W	0*	PCA9685 does not respond to I²C-bus subaddress 3.
			1	PCA9685 responds to I²C-bus subaddress 3.
0	ALLCALL	R/W	0	PCA9685 does not respond to LED All Call I²C-bus address.
			1*	PCA9685 responds to LED All Call I²C-bus address.

[1] When the Auto Increment flag is set, AI = 1, the Control register is automatically incremented after a read or write. This allows the user to program the registers sequentially.

[2] It takes 500 µs max. for the oscillator to be up and running once SLEEP bit has been set to logic 0. Timings on LEDn outputs are not guaranteed if PWM control registers are accessed within the 500 µs window. There is no start-up delay required when using the EXTCLK pin as the PWM clock.

[3] No PWM control is possible when the oscillator is off.

[4] When the oscillator is off (Sleep mode) the LEDn outputs cannot be turned on, off or dimmed/blinked.

7.3.1.1 Restart mode

If the PCA9685 is operating and the user decides to put the chip to sleep (setting MODE1 bit 4) without stopping any of the PWM channels, the RESTART bit (MODE1 bit 7) will be set to logic 1 at the end of the PWM refresh cycle. The contents of each PWM register are held valid when the clock is off.

To restart all of the previously active PWM channels with a few I²C-bus cycles do the following steps:

1. Read MODE1 register.
2. Check that bit 7 (RESTART) is a logic 1. If it is, clear bit 4 (SLEEP). Allow time for oscillator to stabilize (500 μs).
3. Write logic 1 to bit 7 of MODE1 register. All PWM channels will restart and the RESTART bit will clear.

Remark: The SLEEP bit **must** be logic 0 for at least 500 μs, before a logic 1 is written into the RESTART bit.

Other actions that will clear the RESTART bit are:

1. Power cycle.
2. I²C Software Reset command.
3. If the MODE2 OCH bit is logic 0, write to any PWM register then issue an I²C-bus STOP.
4. If the MODE2 OCH bit is logic 1, write to all four PWM registers in any PWM channel.

Likewise, if the user does an orderly shutdown[1] of all the PWM channels before setting the SLEEP bit, the RESTART bit will be cleared. If this is done the contents of all PWM registers are invalidated and must be reloaded before reuse.

An example of the use of the RESTART bit would be the restoring of a customer's laptop LCD backlight intensity coming out of Standby to the level it was before going into Standby.

1. Two methods can be used to do an orderly shutdown. The fastest is to write a logic 1 to bit 4 in register ALL_LED_OFF_H. The other method is to write logic 1 to bit 4 in each active PWM channel LEDn_OFF_H register.

7.3.2 Mode register 2, MODE2

Table 6. MODE2 - Mode register 2 (address 01h) bit description

*Legend: * default value.*

Bit	Symbol	Access	Value	Description
7 to 5	-	read only	000*	reserved
4	INVRT[1]	R/W	0*	Output logic state not inverted. Value to use when external driver used. Applicable when $\overline{OE}$ = 0.
			1	Output logic state inverted. Value to use when no external driver used. Applicable when $\overline{OE}$ = 0.
3	OCH	R/W	0*	Outputs change on STOP command[2].
			1	Outputs change on ACK[3].
2	OUTDRV[1]	R/W	0	The 16 LEDn outputs are configured with an open-drain structure.
			1*	The 16 LEDn outputs are configured with a totem pole structure.
1 to 0	OUTNE[1:0][4]	R/W	00*	When $\overline{OE}$ = 1 (output drivers not enabled), LEDn = 0.
			01	When $\overline{OE}$ = 1 (output drivers not enabled): LEDn = 1 when OUTDRV = 1 LEDn = high-impedance when OUTDRV = 0 (same as OUTNE[1:0] = 10)
			1X	When $\overline{OE}$ = 1 (output drivers not enabled), LEDn = high-impedance.

[1] See Section 7.7 “Using the PCA9685 with and without external drivers” for more details. Normal LEDs can be driven directly in either mode. Some newer LEDs include integrated Zener diodes to limit voltage transients, reduce EMI, protect the LEDs and these must be driven only in the open-drain mode to prevent overheating the IC. Power on reset default state of LEDn output pins is LOW.

[2] Change of the outputs at the STOP command allows synchronizing outputs of more than one PCA9685. Applicable to registers from 06h (LED0_ON_L) to 45h (LED15_OFF_H) only. 1 or more registers can be written, in any order, before STOP.

[3] Update on ACK requires **all** 4 PWM channel registers to be loaded before outputs will change on the **last** ACK.

[4] See Section 7.4 “Active LOW output enable input” for more details.

7.3.3 LED output and PWM control

The turn-on time of each LED driver output and the duty cycle of PWM can be controlled independently using the LEDn_ON and LEDn_OFF registers.

There will be two 12-bit registers per LED output. These registers will be programmed by the user. Both registers will hold a value from 0 to 4095. One 12-bit register will hold a value for the ON time and the other 12-bit register will hold the value for the OFF time. The ON and OFF times are compared with the value of a 12-bit counter that will be running continuously from 0000h to 0FFFh (0 to 4095 decimal).

Update on ACK requires all 4 PWM channel registers to be loaded before outputs will change on the last ACK.

The ON time, which is programmable, will be the time the LED output will be asserted and the OFF time, which is also programmable, will be the time when the LED output will be negated. In this way, the phase shift becomes completely programmable. The resolution for the phase shift is $^1/_{4096}$ of the target frequency. Table 7 lists these registers.

The following two examples illustrate how to calculate values to be loaded into these registers.

Example 1: (assumes that the LED0 output is used and (delay time) + (PWM duty cycle) ≤ 100 %)

Delay time = 10 %; PWM duty cycle = 20 % (LED on time = 20 %; LED off time = 80 %).

Delay time = 10 % = 409.6 ~ 410 counts = 19Ah.

Since the counter starts at 0 and ends at 4095, we will subtract 1, so delay time = 199h counts.

LED0_ON_H = 1h; LED0_ON_L = 99h (LED start turn on after this delay count to 409)

LED on time = 20 % = 819.2 ~ 819 counts.

LED off time = 4CCh (decimal 410 + 819 − 1 = 1228)

LED0_OFF_H = 4h; LED0_OFF_L = CCh (LED start turn off after this count to 1228)

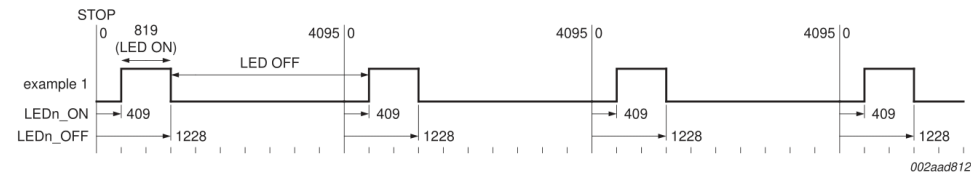


Fig 7. LED output, example 1

Example 2: (assumes that the LED4 output is used and (delay time) + (PWM duty cycle > 100 %)

Delay time = 90 %; PWM duty cycle = 90 % (LED on time = 90 %; LED off time = 10 %).

Delay time = 90 % = 3686.4 ~ 3686 counts − 1 = 3685 = E65h.

LED4_ON_H = Eh; LED4_ON_L = 65h (LED start turn on after this delay count to 3685)

LED on time = 90 % = 3686 counts.

Since the delay time and LED on period of the duty cycle is greater than 4096 counts, the LEDn_OFF count will occur in the next frame. Therefore, 4096 is subtracted from the LEDn_OFF count to get the correct LEDn_OFF count. See Figure 9, Figure 10 and Figure 11.

LED off time = CCBh (decimal 3685 + 3686 = 7372 − 4096 = 3275)

LED4_OFF_H = Ch; LED4_OFF_L = CBh (LED start turn off after this count to 3275)

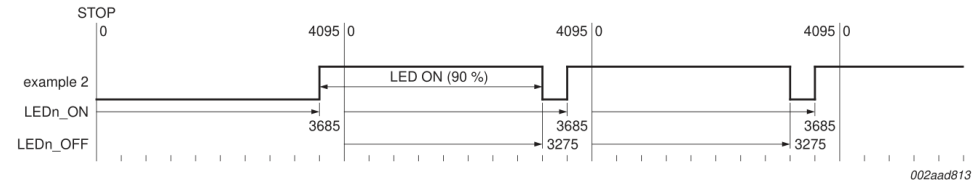
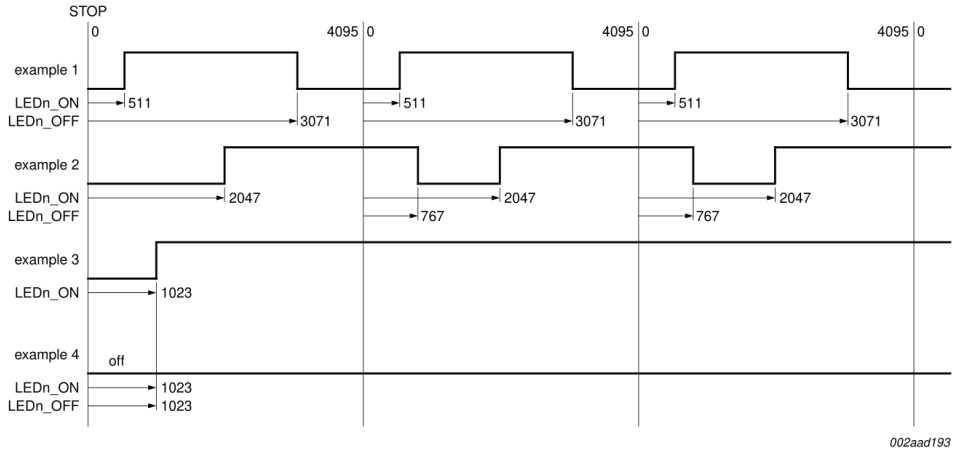


Fig 8. LED output, example 2



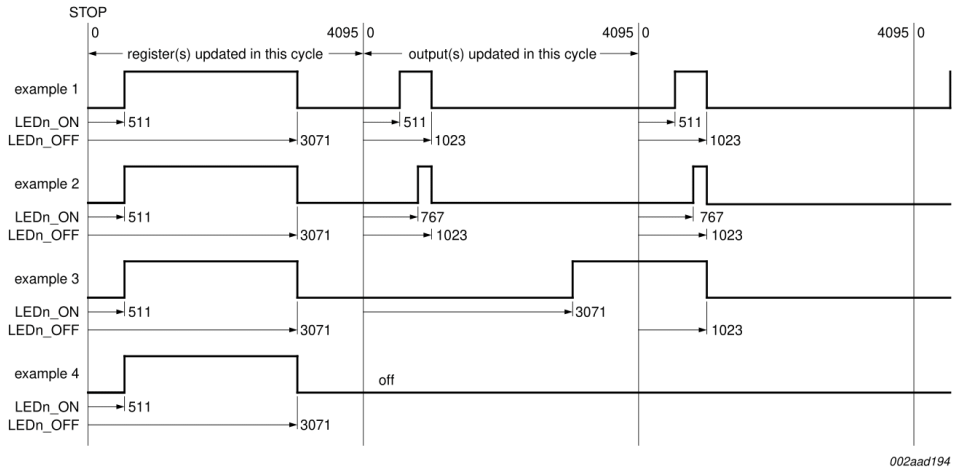
Example 1: LEDn_ON (511) < LEDn_OFF (3071)

Example 2: LEDn_ON (2047) > LEDn_OFF (767)

Example 3: LEDn_ON[12] = 1; LEDn_ON[11:0] = 1022; LEDn_OFF[12] = 0; LEDn_OFF[11:0] = don't care

Example 4: LEDn_ON[12] = 0; LEDn_OFF[12] = 0; LEDn_ON[11:0] = LEDn_OFF[11:0]

Fig 9. Output example



Example 1: LEDn_ON unchanged and LEDn_OFF decreased.

Example 2: LEDn_ON increased and LEDn_OFF decreased.

Example 3: LEDn_ON made > LEDn_OFF.

Example 4: LEDn_OFF[12] set to 1.

Fig 10. Update examples when LEDn_ON < LEDn_OFF

Example 1: LEDn_ON unchanged and LEDn_OFF decreased, but delay still > LEDn_OFF
Example 2: LEDn_ON changed and LEDn_OFF changed, but delay still > LEDn_OFF
Example 3: LEDn_ON unchanged and LEDn_OFF increased where LEDn_ON < LEDn_OFF
Example 4: LEDn_ON[12] = 1 and LEDn_OFF[12] changed from 0 to 1

Fig 11. Update examples when LEDn_ON > LEDn_OFF

Table 7. LED_ON, LED_OFF control registers (address 06h to 45h) bit description

*Legend: * default value.*

Address	Register	Bit	Symbol	Access	Value	Description
06h	LED0_ON_L	7:0	LED0_ON_L[7:0]	R/W	0000 0000*	LEDn_ON count for LED0, 8 LSBs
07h	LED0_ON_H	7:5	reserved	R	000*	non-writable
		4	LED0_ON_H[4]	R/W	0*	LED0 full ON
		3:0	LED0_ON_H[3:0]	R/W	0000*	LEDn_ON count for LED0, 4 MSBs
08h	LED0_OFF_L	7:0	LED0_OFF_L[7:0]	R/W	0000 0000*	LEDn_OFF count for LED0, 8 LSBs
09h	LED0_OFF_H	7:5	reserved	R	000*	non-writable
		4	LED0_OFF_H[4]	R/W	1*	LED0 full OFF
		3:0	LED0_OFF_H[3:0]	R/W	0000*	
0Ah	LED1_ON_L	7:0	LED1_ON_L[7:0]	R/W	0000 0000*	LEDn_ON count for LED1, 8 LSBs
0Bh	LED1_ON_H	7:5	reserved	R	000*	non-writable
		4	LED1_ON_H[4]	R/W	0*	LED1 full ON
		3:0	LED1_ON_H[3:0]	R/W	0000*	LEDn_ON count for LED1, 4 MSBs
0Ch	LED1_OFF_L	7:0	LED1_OFF_L[7:0]	R/W	0000 0000*	LEDn_OFF count for LED1, 8 LSBs
0Dh	LED1_OFF_H	7:5	reserved	R	000*	non-writable
		4	LED1_OFF_H[4]	R/W	1*	LED1 full OFF
		3:0	LED1_OFF_H[3:0]	R/W	0000*	LEDn_OFF count for LED1, 4 MSBs
0Eh	LED2_ON_L	7:0	LED2_ON_L[7:0]	R/W	0000 0000*	LEDn_ON count for LED2, 8 LSBs
0Fh	LED2_ON_H	7:5	reserved	R	000*	non-writable
		4	LED2_ON_H[4]	R/W	0*	LED2 full ON
		3:0	LED2_ON_H[3:0]	R/W	0000*	LEDn_ON count for LED2, 4 MSBs
10h	LED2_OFF_L	7:0	LED2_OFF_L[7:0]	R/W	0000 0000*	LEDn_OFF count for LED2, 8 LSBs
11h	LED2_OFF_H	7:5	reserved	R	000*	non-writable
		4	LED2_OFF_H[4]	R/W	1*	LED2 full OFF
		3:0	LED2_OFF_H[3:0]	R/W	0000*	LEDn_OFF count for LED2, 4 MSBs
12h	LED3_ON_L	7:0	LED3_ON_L[7:0]	R/W	0000 0000*	LEDn_ON count for LED3, 8 LSBs
13h	LED3_ON_H	7:5	reserved	R	000*	non-writable
		4	LED3_ON_H[4]	R/W	0*	LED3 full ON
		3:0	LED3_ON_H[3:0]	R/W	0000*	LEDn_ON count for LED3, 4 MSBs
14h	LED3_OFF_L	7:0	LED3_OFF_L[7:0]	R/W	0000 0000*	LEDn_OFF count for LED3, 8 LSBs
15h	LED3_OFF_H	7:5	reserved	R	000*	non-writable
		4	LED3_OFF_H[4]	R/W	1*	LED3 full OFF
		3:0	LED3_OFF_H[3:0]	R/W	0000*	LEDn_OFF count for LED3, 4 MSBs
16h	LED4_ON_L	7:0	LED4_ON_L[7:0]	R/W	0000 0000*	LEDn_ON count for LED4, 8 LSBs
17h	LED4_ON_H	7:5	reserved	R	000*	non-writable
		4	LED4_ON_H[4]	R/W	0*	LED4 full ON
		3:0	LED4_ON_H[3:0]	R/W	0000*	LEDn_ON count for LED4, 4 MSBs

Table 7. LED_ON, LED_OFF control registers (address 06h to 45h) bit description *...continued*

*Legend: * default value.*

Address	Register	Bit	Symbol	Access	Value	Description
18h	LED4_OFF_L	7:0	LED4_OFF_L[7:0]	R/W	0000 0000*	LEDn_OFF count for LED4, 8 LSBs
19h	LED4_OFF_H	7:5	reserved	R	000*	non-writable
		4	LED4_OFF_H[4]	R/W	1*	LED4 full OFF
		3:0	LED4_OFF_H[3:0]	R/W	0000*	LEDn_OFF count for LED4, 4 MSBs
1Ah	LED5_ON_L	7:0	LED5_ON_L[7:0]	R/W	0000 0000*	LEDn_ON count for LED5, 8 LSBs
1Bh	LED5_ON_H	7:5	reserved	R	000*	non-writable
		4	LED5_ON_H[4]	R/W	0*	LED5 full ON
		3:0	LED5_ON_H[3:0]	R/W	0000*	LEDn_ON count for LED5, 4 MSBs
1Ch	LED5_OFF_L	7:0	LED5_OFF_L[7:0]	R/W	0000 0000*	LEDn_OFF count for LED5, 8 LSBs
1Dh	LED5_OFF_H	7:5	reserved	R	000*	non-writable
		4	LED5_OFF_H[4]	R/W	1*	LED5 full OFF
		3:0	LED5_OFF_H[3:0]	R/W	0000*	LEDn_OFF count for LED5, 4 MSBs
1Eh	LED6_ON_L	7:0	LED6_ON_L[7:0]	R/W	0000 0000*	LEDn_ON count for LED6, 8 LSBs
1Fh	LED6_ON_H	7:5	reserved	R	000*	non-writable
		4	LED6_ON_H[4]	R/W	0*	LED6 full ON
		3:0	LED6_ON_H[3:0]	R/W	0000*	LEDn_ON count for LED6, 4 MSBs
20h	LED6_OFF_L	7:0	LED6_OFF_L[7:0]	R/W	0000 0000*	LEDn_OFF count for LED6, 8 LSBs
21h	LED6_OFF_H	7:5	reserved	R	000*	non-writable
		4	LED6_OFF_H[4]	R/W	1*	LED6 full OFF
		3:0	LED6_OFF_H[3:0]	R/W	0000*	LEDn_OFF count for LED6, 4 MSBs
22h	LED7_ON_L	7:0	LED7_ON_L[7:0]	R/W	0000 0000*	LEDn_ON count for LED7, 8 LSBs
23h	LED7_ON_H	7:5	reserved	R	000*	non-writable
		4	LED7_ON_H[4]	R/W	0*	LED7 full ON
		3:0	LED7_ON_H[3:0]	R/W	0000*	LEDn_ON count for LED7, 4 MSBs
24h	LED7_OFF_L	7:0	LED7_OFF_L[7:0]	R/W	0000 0000*	LEDn_OFF count for LED7, 8 LSBs
25h	LED7_OFF_H	7:5	reserved	R	000*	non-writable
		4	LED7_OFF_H[4]	R/W	1*	LED7 full OFF
		3:0	LED7_OFF_H[3:0]	R/W	0000*	LEDn_OFF count for LED7, 4 MSBs
26h	LED8_ON_L	7:0	LED8_ON_L[7:0]	R/W	0000 0000*	LEDn_ON count for LED8, 8 LSBs
27h	LED8_ON_H	7:5	reserved	R	000*	non-writable
		4	LED8_ON_H[4]	R/W	0*	LED8 full ON
		3:0	LED8_ON_H[3:0]	R/W	0000*	LEDn_ON count for LED8, 4 MSBs
28h	LED8_OFF_L	7:0	LED8_OFF_L[7:0]	R/W	0000 0000*	LEDn_OFF count for LED8, 8 LSBs
29h	LED8_OFF_H	7:5	reserved	R	000*	non-writable
		4	LED8_OFF_H[4]	R/W	1*	LED8 full OFF
		3:0	LED8_OFF_H[3:0]	R/W	0000*	LEDn_OFF count for LED8, 4 MSBs

Table 7. LED_ON, LED_OFF control registers (address 06h to 45h) bit description *...continued*

*Legend: * default value.*

Address	Register	Bit	Symbol	Access	Value	Description
2Ah	LED9_ON_L	7:0	LED9_ON_L[7:0]	R/W	0000 0000*	LEDn_ON count for LED9, 8 LSBs
2Bh	LED9_ON_H	7:5	reserved	R	000*	non-writable
		4	LED9_ON_H[4]	R/W	0*	LED9 full ON
		3:0	LED9_ON_H[3:0]	R/W	0000*	LEDn_ON count for LED9, 4 MSBs
2Ch	LED9_OFF_L	7:0	LED9_OFF_L[7:0]	R/W	0000 0000*	LEDn_OFF count for LED9, 8 LSBs
2Dh	LED9_OFF_H	7:5	reserved	R	000*	non-writable
		4	LED9_OFF_H[4]	R/W	1*	LED9 full OFF
		3:0	LED9_OFF_H[3:0]	R/W	0000*	LEDn_OFF count for LED9, 4 MSBs
2Eh	LED10_ON_L	7:0	LED10_ON_L[7:0]	R/W	0000 0000*	LEDn_ON count for LED10, 8 LSBs
2Fh	LED10_ON_H	7:5	reserved	R	000*	non-writable
		4	LED10_ON_H[4]	R/W	0*	LED10 full ON
		3:0	LED10_ON_H[3:0]	R/W	0000*	LEDn_ON count for LED10, 4 MSBs
30h	LED10_OFF_L	7:0	LED10_OFF_L[7:0]	R/W	0000 0000*	LEDn_OFF count for LED10, 8 LSBs
31h	LED10_OFF_H	7:5	reserved	R	000*	non-writable
		4	LED10_OFF_H[4]	R/W	1*	LED10 full OFF
		3:0	LED10_OFF_H[3:0]	R/W	0000*	LEDn_OFF count for LED10, 4 MSBs
32h	LED11_ON_L	7:0	LED11_ON_L[7:0]	R/W	0000 0000*	LEDn_ON count for LED11, 8 LSBs
33h	LED11_ON_H	7:5	reserved	R	000*	non-writable
		4	LED11_ON_H[4]	R/W	0*	LED11 full ON
		3:0	LED11_ON_H[3:0]	R/W	0000*	LEDn_ON count for LED11, 4 MSBs
34h	LED11_OFF_L	7:0	LED11_OFF_L[7:0]	R/W	0000 0000*	LEDn_OFF count for LED11, 8 LSBs
35h	LED11_OFF_H	7:5	reserved	R	000*	non-writable
		4	LED11_OFF_H[4]	R/W	1*	LED11 full OFF
		3:0	LED11_OFF_H[3:0]	R/W	0000*	LEDn_OFF count for LED11, 4 MSBs
36h	LED12_ON_L	7:0	LED12_ON_L[7:0]	R/W	0000 0000*	LEDn_ON count for LED12, 8 LSBs
37h	LED12_ON_H	7:5	reserved	R	000*	non-writable
		4	LED12_ON_H[4]	R/W	0*	LED12 full ON
		3:0	LED12_ON_H[3:0]	R/W	0000*	LEDn_ON count for LED12, 4 MSBs
38h	LED12_OFF_L	7:0	LED12_OFF_L[7:0]	R/W	0000 0000*	LEDn_OFF count for LED12, 8 LSBs
39h	LED12_OFF_H	7:5	reserved	R	000*	non-writable
		4	LED12_OFF_H[4]	R/W	1*	LED12 full OFF
		3:0	LED12_OFF_H[3:0]	R/W	0000*	LEDn_OFF count for LED12, 4 MSBs
3Ah	LED13_ON_L	7:0	LED13_ON_L[7:0]	R/W	0000 0000*	LEDn_ON count for LED13, 8 LSBs
3Bh	LED13_ON_H	7:5	reserved	R	000*	non-writable
		4	LED13_ON_H[4]	R/W	0*	LED13 full ON
		3:0	LED13_ON_H[3:0]	R/W	0000*	LEDn_ON count for LED13, 4 MSBs

Table 7. LED_ON, LED_OFF control registers (address 06h to 45h) bit description *...continued*

*Legend: * default value.*

Address	Register	Bit	Symbol	Access	Value	Description
3Ch	LED13_OFF_L	7:0	LED13_OFF_L[7:0]	R/W	0000 0000*	LEDn_OFF count for LED13, 8 LSBs
3Dh	LED13_OFF_H	7:5	reserved	R	000*	non-writable
		4	LED13_OFF_H[4]	R/W	1*	LED13 full OFF
		3:0	LED13_OFF_H[3:0]	R/W	0000*	LEDn_OFF count for LED13, 4 MSBs
3Eh	LED14_ON_L	7:0	LED14_ON_L[7:0]	R/W	0000 0000*	LEDn_ON count for LED14, 8 LSBs
3Fh	LED14_ON_H	7:5	reserved	R	000*	non-writable
		4	LED14_ON_H[4]	R/W	0*	LED14 full ON
		3:0	LED14_ON_H[3:0]	R/W	0000*	LEDn_ON count for LED14, 4 MSBs
40h	LED14_OFF_L	7:0	LED14_OFF_L[7:0]	R/W	0000 0000*	LEDn_OFF count for LED14, 8 LSBs
41h	LED14_OFF_H	7:5	reserved	R	000*	non-writable
		4	LED14_OFF_H[4]	R/W	1*	LED14 full OFF
		3:0	LED14_OFF_H[3:0]	R/W	0000*	LEDn_OFF count for LED14, 4 MSBs
42h	LED15_ON_L	7:0	LED15_ON_L[7:0]	R/W	0000 0000*	LEDn_ON count for LED15, 8 LSBs
43h	LED15_ON_H	7:5	reserved	R	000*	non-writable
		4	LED15_ON_H[4]	R/W	0*	LED15 full ON
		3:0	LED15_ON_H[3:0]	R/W	0000*	LEDn_ON count for LED15, 4 MSBs
44h	LED15_OFF_L	7:0	LED15_OFF_L[7:0]	R/W	0000 0000*	LEDn_OFF count for LED15, 8 LSBs
45h	LED15_OFF_H	7:5	reserved	R	000*	non-writable
		4	LED15_OFF_H[4]	R/W	1*	LED15 full OFF
		3:0	LED15_OFF_H[3:0]	R/W	0000*	LEDn_OFF count for LED15, 4 MSBs

The LEDn_ON_H output control bit 4, when set to logic 1, causes the output to be always ON. The turning ON of the LED is delayed by the amount in the LEDn_ON registers. LEDn_OFF[11:0] are ignored. When this bit = 0, then the LEDn_ON and LEDn_OFF registers are used according to their normal definition.

The LEDn_OFF_H output control bit 4, when set to logic 1, causes the output to be always OFF. In this case the values in the LEDn_ON registers are ignored.

Remark: When all LED outputs are configured as 'always OFF', the prescale counter and all associated PWM cycle timing logic are disabled. If LEDn_ON_H[4] and LEDn_OFF_H[4] are set at the same time, the LEDn_OFF_H[4] function takes precedence.

7.3.4 ALL_LED_ON and ALL_LED_OFF control

The ALL_LED_ON and ALL_LED_OFF registers allow just four I²C-bus write sequences to fill all the ON and OFF registers with the same patterns.

Table 8. ALL_LED_ON and ALL_LED_OFF control registers (address FAh to FEh) bit description

*Legend: * default value.*

Address	Register	Bit	Symbol	Access	Value	Description
FAh	ALL_LED_ON_L	7:0	ALL_LED_ON_L[7:0]	W only	0000 0000*	LEDn_ON count for ALL_LED, 8 MSBs
FBh	ALL_LED_ON_H	7:5	reserved	R	000*	non-writable
		4	ALL_LED_ON_H[4]	W only	1*	ALL_LED full ON
		3:0	ALL_LED_ON_H[3:0]	W only	0000*	LEDn_ON count for ALL_LED, 4 MSBs
FCh	ALL_LED_OFF_L	7:0	ALL_LED_OFF_L[7:0]	W only	0000 0000*	LEDn_OFF count for ALL_LED, 8 MSBs
FDh	ALL_LED_OFF_H	7:5	reserved	R	000*	non-writable
		4	ALL_LED_OFF_H[4]	W only	1*	ALL_LED full OFF
		3:0	ALL_LED_OFF_H[3:0]	W only	0000*	LEDn_OFF count for ALL_LED, 4 MSBs
FEh	PRE_SCALE	7:0	PRE_SCALE[7:0]	R/W	0001 1110*	prescaler to program the PWM output frequency (default is 200 Hz)

The LEDn_ON and LEDn_OFF counts can vary from 0 to 4095. The LEDn_ON and LEDn_OFF count registers should never be programmed with the same values.

Because the loading of the LEDn_ON and LEDn_OFF registers is via the I²C-bus, and asynchronous to the internal oscillator, we want to ensure that we do not see any visual artifacts of changing the ON and OFF values. This is achieved by updating the changes at the end of the LOW cycle.

7.3.5 PWM frequency PRE_SCALE

The hardware forces a minimum value that can be loaded into the PRE_SCALE register at '3'. The PRE_SCALE register defines the frequency at which the outputs modulate. The prescale value is determined with the formula shown in Equation 1:

$$prescale\ value = round\left(\frac{osc\_clock}{4096 \times update\_rate}\right) - 1 \quad (1)$$

where the update rate is the output modulation frequency required. For example, for an output default frequency of 200 Hz with an oscillator clock frequency of 25 MHz:

$$prescale\ value = round\left(\frac{25\ MHz}{4096 \times 200}\right) - 1 = 30\ (0x1Eh) \quad (2)$$

The maximum PWM frequency is 1526 Hz if the PRE_SCALE register is set "0x03h".

The minimum PWM frequency is 24 Hz if the PRE_SCALE register is set "0xFFh".

The PRE_SCALE register can only be set when the SLEEP bit of MODE1 register is set to logic 1.

7.3.6 SUBADR1 to SUBADR3, I²C-bus subaddress 1 to 3

Table 9. SUBADR1 to SUBADR3 - I²C-bus subaddress registers 0 to 3 (address 02h to 04h) bit description

*Legend: * default value.*

Address	Register	Bit	Symbol	Access	Value	Description
02h	SUBADR1	7:1	A1[7:1]	R/W	1110 001*	I²C-bus subaddress 1
		0	A1[0]	R only	0*	reserved
03h	SUBADR2	7:1	A2[7:1]	R/W	1110 010*	I²C-bus subaddress 2
		0	A2[0]	R only	0*	reserved
04h	SUBADR3	7:1	A3[7:1]	R/W	1110 100*	I²C-bus subaddress 3
		0	A3[0]	R only	0*	reserved

Subaddresses are programmable through the I²C-bus. Default power-up values are E2h, E4h, E8h, and the device(s) will not acknowledge these addresses right after power-up (the corresponding SUBx bit in MODE1 register is equal to 0).

Once subaddresses have been programmed to their right values, SUBx bits need to be set to logic 1 in order to have the device acknowledging these addresses (MODE1 register).

Only the 7 MSBs representing the I²C-bus subaddress are valid. The LSB in SUBADRx register is a read-only bit (0).

When SUBx is set to logic 1, the corresponding I²C-bus subaddress can be used during either an I²C-bus read or write sequence.

7.3.7 ALLCALLADR, LED All Call I²C-bus address

Table 10. ALLCALLADR - LED All Call I²C-bus address register (address 05h) bit description

*Legend: * default value.*

Address	Register	Bit	Symbol	Access	Value	Description
05h	ALLCALLADR	7:1	AC[7:1]	R/W	1110 000*	ALLCALL I²C-bus address register
		0	AC[0]	R only	0*	reserved

The LED All Call I²C-bus address allows all the PCA9685s in the bus to be programmed at the same time (ALLCALL bit in register MODE1 must be equal to 1 (power-up default state)). This address is programmable through the I²C-bus and can be used during either an I²C-bus read or write sequence. The register address can also be programmed as a Sub Call.

Only the 7 MSBs representing the All Call I²C-bus address are valid. The LSB in ALLCALLADR register is a read-only bit (0).

If ALLCALL bit = 0, the device does not acknowledge the address programmed in register ALLCALLADR.

7.4 Active LOW output enable input

The active LOW output enable ($\overline{OE}$) pin, allows to enable or disable all the LED outputs at the same time.

- When a LOW level is applied to $\overline{OE}$ pin, all the LED outputs are enabled and follow the output state defined in the LEDn_ON and LEDn_OFF registers with the polarity defined by INVRT bit (MODE2 register).
- When a HIGH level is applied to $\overline{OE}$ pin, all the LED outputs are programmed to the value that is defined by OUTNE[1:0] in the MODE2 register.

Table 11. LED outputs when $\overline{OE}$ = 1

OUTNE1	OUTNE0	LED outputs
0	0	0
0	1	1 if OUTDRV = 1, high-impedance if OUTDRV = 0
1	0	high-impedance
1	1	high-impedance

The $\overline{OE}$ pin can be used as a synchronization signal to switch on/off several PCA9685 devices at the same time. This requires an external clock reference that provides blinking period and the duty cycle.

The $\overline{OE}$ pin can also be used as an external dimming control signal. The frequency of the external clock must be high enough not to be seen by the human eye, and the duty cycle value determines the brightness of the LEDs.

7.5 Power-on reset

When power is applied to V_{DD}, an internal power-on reset holds the PCA9685 in a reset condition until V_{DD} has reached V_{POR}. At this point, the reset condition is released and the PCA9685 registers and I²C-bus state machine are initialized to their default states. Thereafter, V_{DD} must be lowered below 0.2 V to reset the device.

7.6 Software reset

The Software Reset Call (SWRST Call) allows all the devices in the I²C-bus to be reset to the power-up state value through a specific formatted I²C-bus command. To be performed correctly, it implies that the I²C-bus is functional and that there is no device hanging the bus.

The SWRST Call function is defined as the following:

1. A START command is sent by the I²C-bus master.
2. The reserved SWRST I²C-bus address '0000 000' with the R/$\overline{W}$ bit set to '0' (write) is sent by the I²C-bus master.
3. The PCA9685 device(s) acknowledge(s) after seeing the General Call address '0000 0000' (00h) only. If the R/$\overline{W}$ bit is set to '1' (read), no acknowledge is returned to the I²C-bus master.
4. Once the General Call address has been sent and acknowledged, the master sends 1 byte with 1 specific value (SWRST data byte 1):
   a. Byte 1 = 06h: the PCA9685 acknowledges this value only. If byte 1 is not equal to 06h, the PCA9685 does not acknowledge it.

   If more than 1 byte of data is sent, the PCA9685 does not acknowledge any more.
5. Once the correct byte (SWRST data byte 1) has been sent and correctly acknowledged, the master sends a STOP command to end the SWRST Call: the PCA9685 then resets to the default value (power-up value) and is ready to be addressed again within the specified bus free time (t_{BUF}).

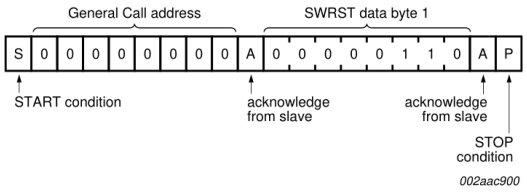


Fig 12. SWRST Call

The I²C-bus master must interpret a non-acknowledge from the PCA9685 (at any time) as a 'SWRST Call Abort'. The PCA9685 does not initiate a reset of its registers. This happens only when the format of the SWRST Call sequence is not correct.

7.7 Using the PCA9685 with and without external drivers

The PCA9685 LED output drivers are 5.5 V only tolerant and can sink up to 25 mA at 5 V.

If the device needs to drive LEDs to a higher voltage and/or higher current, use of an external driver is required.

- INVRT bit (MODE2 register) can be used to keep the LED PWM control firmware the same independently of the type of external driver. This bit allows LED output polarity inversion/non-inversion only when $\overline{OE}$ = 0.
- OUTDRV bit (MODE2 register) allows minimizing the amount of external components required to control the external driver (N-type or P-type device).

Table 12. Use of INVRT and OUTDRV based on connection to the LEDn outputs when $\overline{OE}$ = 0[1]

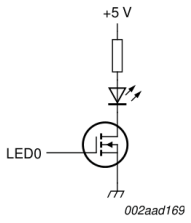
INVRT	OUTDRV	Direct connection to LEDn		External N-type driver		External P-type driver	
		Firmware	External pull-up resistor	Firmware	External pull-up resistor	Firmware	External pull-up resistor
0	0	formulas and LED output state values inverted	LED current limiting R[2]	formulas and LED output state values inverted	required	formulas and LED output state values apply	required
0	1	formulas and LED output state values inverted	LED current limiting R[2]	formulas and LED output state values apply[3]	not required[3]	formulas and LED output state values inverted	not required
1	0	formulas and LED output state values apply[2]	LED current limiting R	formulas and LED output state values apply	required	formulas and LED output state values inverted	required
1	1	formulas and LED output state values apply[2]	LED current limiting R	formulas and LED output state values inverted	not required	formulas and LED output state values apply[4]	not required[4]

[1] When $\overline{OE}$ = 1, LED output state is controlled only by OUTNE[1:0] bits (MODE2 register).

[2] Correct configuration when LEDs directly connected to the LEDn outputs (connection to V_{DD} through current limiting resistor).

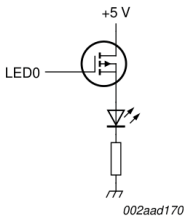
[3] Optimum configuration when external N-type (NPN, NMOS) driver used.

[4] Optimum configuration when external P-type (PNP, PMOS) driver used.



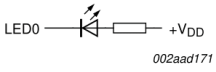
INVRT = 0
OUTDRV = 1

Fig 13. External N-type driver



INVRT = 1
OUTDRV = 1

Fig 14. External P-type driver



INVRT = 1
OUTDRV = 0

Fig 15. Direct LED connection

8. Characteristics of the I²C-bus

The I²C-bus is for 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device. Data transfer may be initiated only when the bus is not busy.

8.1 Bit transfer

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as control signals (see Figure 16).

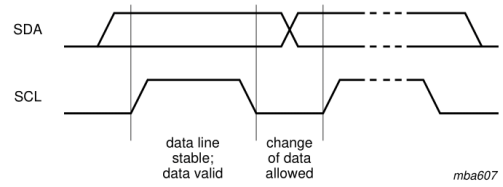


Fig 16. Bit transfer

8.1.1 START and STOP conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line while the clock is HIGH is defined as the START condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the STOP condition (P) (see Figure 17).

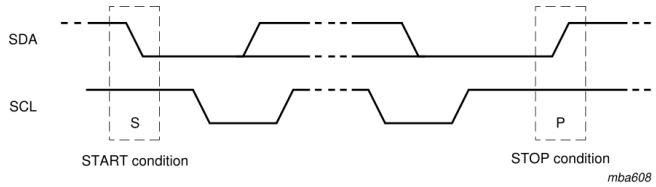


Fig 17. Definition of START and STOP conditions

8.2 System configuration

A device generating a message is a ‘transmitter’; a device receiving is the ‘receiver’. The device that controls the message is the ‘master’ and the devices which are controlled by the master are the ‘slaves’ (see Figure 18).

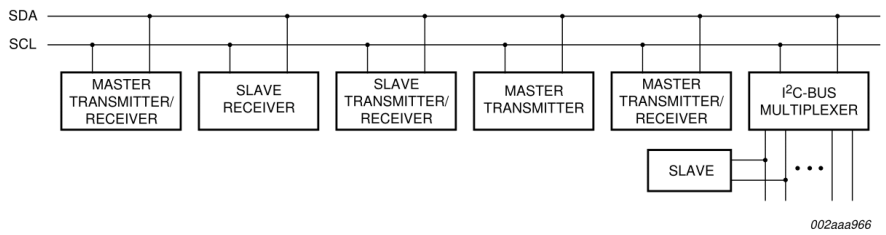


Fig 18. System configuration

8.3 Acknowledge

The number of data bytes transferred between the START and the STOP conditions from transmitter to receiver is not limited. Each byte of eight bits is followed by one acknowledge bit. The acknowledge bit is a HIGH level put on the bus by the transmitter, whereas the master generates an extra acknowledge related clock pulse.

A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse; set-up time and hold time must be taken into account.

A master receiver must signal an end of data to the transmitter by not generating an acknowledge on the last byte that has been clocked out of the slave. In this event, the transmitter must leave the data line HIGH to enable the master to generate a STOP condition.

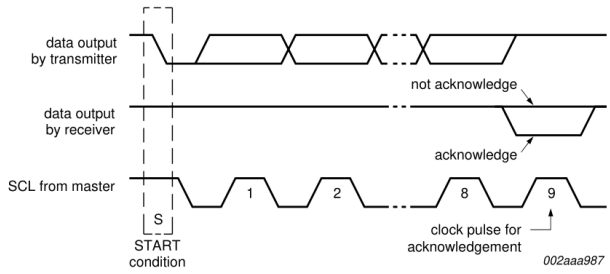
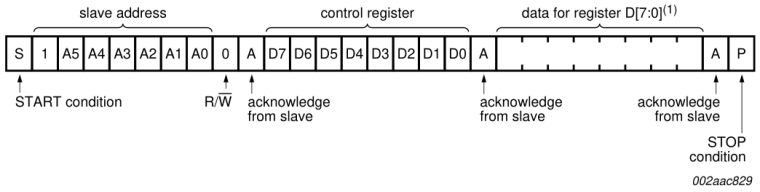


Fig 19. Acknowledgement on the I²C-bus

9. Bus transactions



(1) See Table 4 for register definition.

Fig 20. Write to a specific register

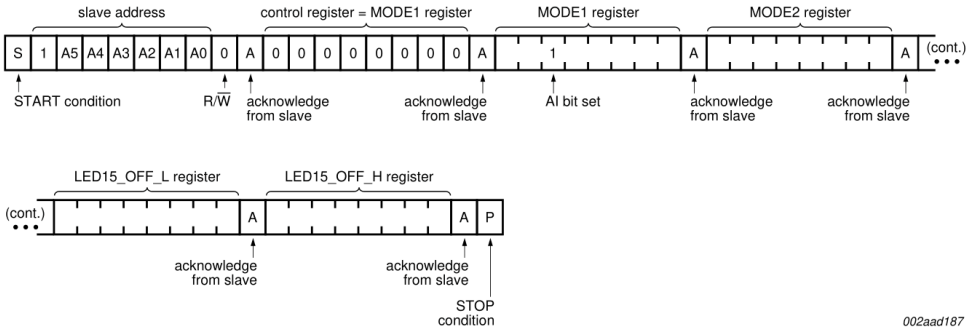


Fig 21. Write to all registers using the Auto-Increment feature; AI initially clear

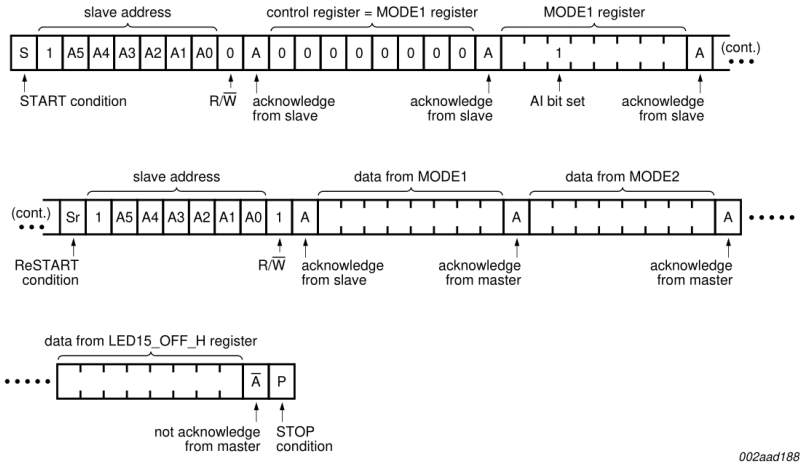


Fig 22. Read all registers using the Auto-Increment feature; AI initially clear

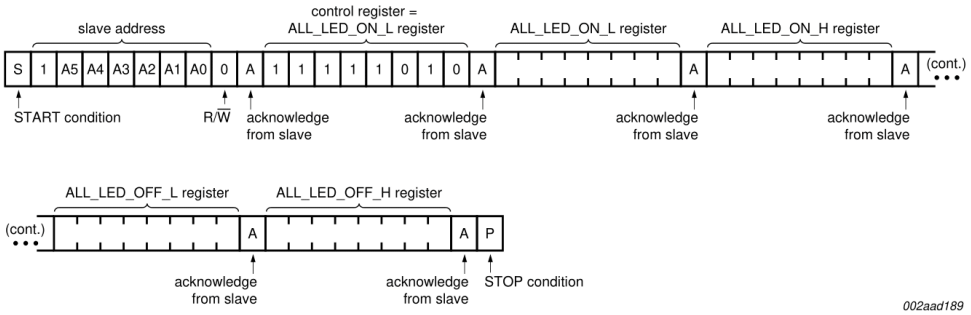


Fig 23. Write to ALL_LED_ON and ALL_LED_OFF registers using the Auto-Increment feature; AI initially set

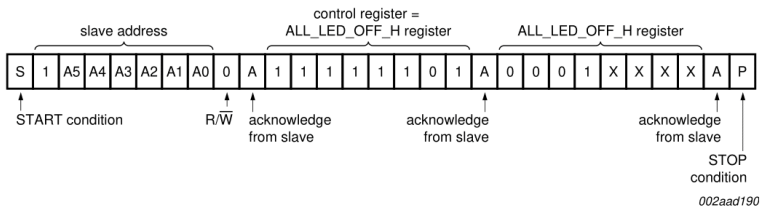
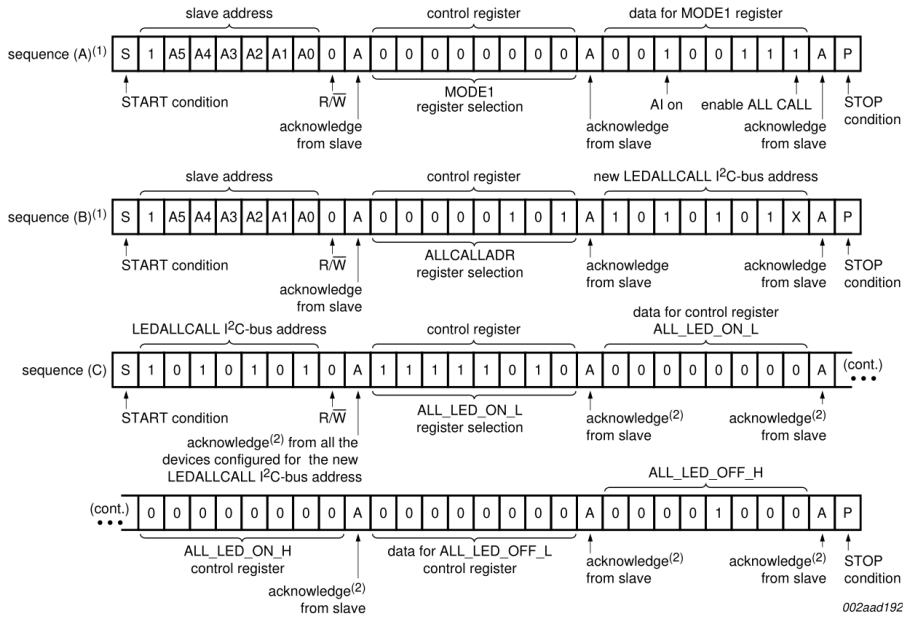


Fig 24. Write to ALL_LED_OFF_H to turn OFF all PWMs

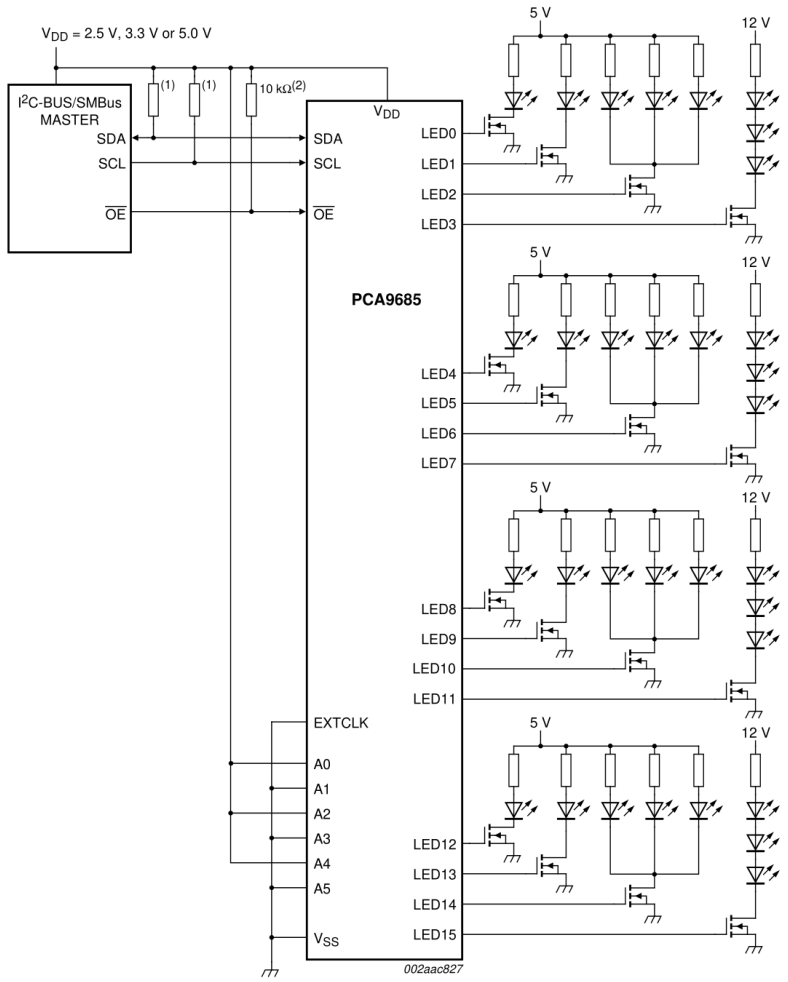


(1) In this example, several PCA9685s are used and the same sequences (A) and (B) above are sent to each of them.

(2) Acknowledge from all the slave devices configured for the new LED All Call I²C-bus address in sequence (B).

Fig 25. LED All Call I²C-bus address programming and LED All Call sequence example

10. Application design-in information



I²C-bus address = 1010 101x.

All 16 of the LEDn outputs configurable as either open-drain or totem pole. Mixing of configuration is not possible.

Remark: Set INVRT = 0, OUTDRV = 1, OUTNE = 01 (MODE2 register bits)

(1) Resistor value should be chosen by referencing section 7 of *UM10204, "I²C-bus specification and user manual"*.

(2) $\overline{OE}$ requires pull-up resistor if control signal from the master is open-drain.

Fig 26. Typical application

Question 1: What kind of edge rate control is there on the outputs?

- The typical edge rates depend on the output configuration, supply voltage, and the applied load. The outputs can be configured as either open-drain NMOS or totem pole outputs. If the customer is using the part to directly drive LEDs, they should be using it in an open-drain NMOS, if they are concerned about the maximum I_{SS} and ground bounce. The edge rate control was designed primarily to slow down the turn-on of the output device; it turns off rather quickly (~1.5 ns). In simulation, the typical turn-on time for the open-drain NMOS was ~14 ns (V_{DD} = 3.6 V; C_L = 50 pF; R_{PU} = 500 Ω).

Question 2: Is ground bounce possible?

- Ground bounce is a possibility, especially if all 16 outputs are changed at full current (25 mA each). There is a fair amount of decoupling capacitance on chip (~50 pF), which is intended to suppress some of the ground bounce. The customer will need to determine if additional decoupling capacitance externally placed as close as physically possible to the device is required.

Question 3: Can I really sink 400 mA through the single ground pin on the package and will this cause any ground bounce problem due to the PWM of the LEDs?

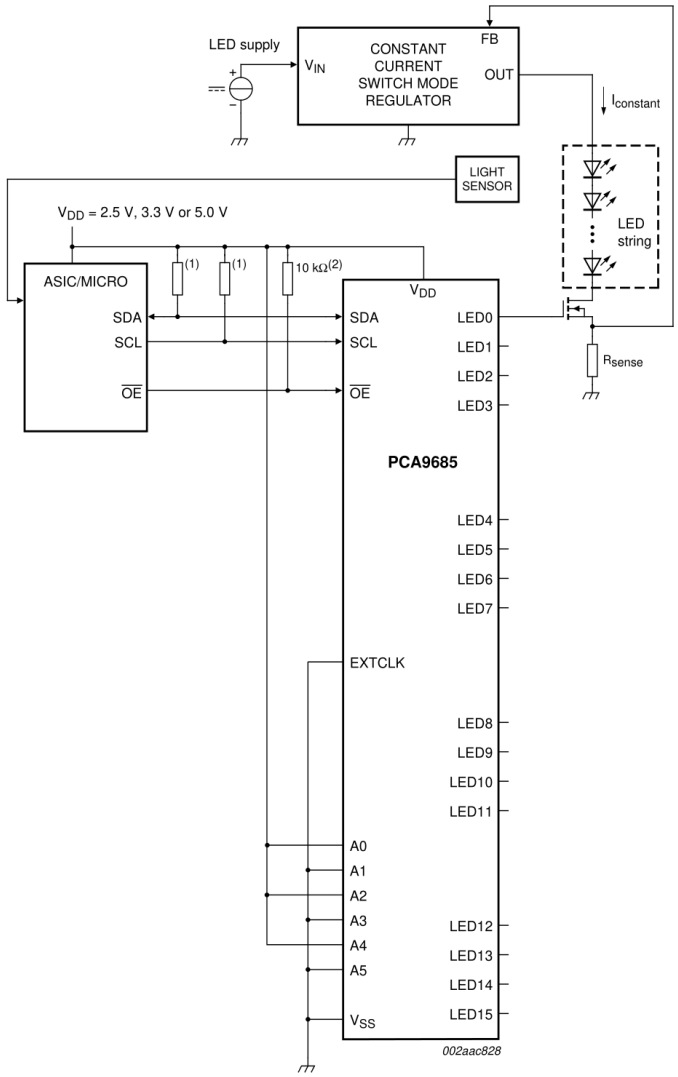
- Yes, you can sink 400 mA through a single ground pin on the **package**. Although the package only has one ground pin, there are two ground pads on the die itself connected to this one pin. Although some ground bounce is likely, it will not disrupt the operation of the part and would be reduced by the external decoupling capacitance.

Question 4: I can't turn the LEDs on or off, but their registers are set properly. Why?

- Check the MODE1 register SLEEP (bit 4) setting. The bit needs to be 0 in order to enable the clocking. If both clock sources (internal osc and EXTCLK) are turned OFF (bit 4 = 1), the LEDs cannot be dimmed or blinked.

Question 5: I'm using LEDs with integrated Zener diodes and the IC is getting very hot. Why?

- The IC outputs can be set to either open-drain or push-pull and default to push-pull outputs. In this application with the Zener diodes, they need to be set to open-drain since in the push-pull architecture there is a low resistance path to GND through the Zener and this is causing the IC to overheat.



I²C-bus address = 1010 101x.

Remark: Set INVRT = 0, OUTDRV = 1, OUTNE = 01 (MODE2 register bits) for this configuration.

(1) Resistor value should be chosen by referencing Section 7 of *UM10204, "I²C-bus specification and user manual"*.

(2) $\overline{OE}$ requires pull-up resistor if control signal from the master is open-drain.

Fig 27. LCD backlighting application

11. Limiting values

Table 13. Limiting values

In accordance with the Absolute Maximum Rating System (IEC 60134).

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DD}	supply voltage		−0.5	+6.0	V
$V_{I/O}$	voltage on an input/output pin		V_{SS} − 0.5	5.5	V
$I_{O(LEDn)}$	output current on pin LEDn		-	25	mA
I_{SS}	ground supply current		-	400	mA
P_{tot}	total power dissipation		-	400	mW
T_{stg}	storage temperature		−65	+150	°C
T_{amb}	ambient temperature	operating	−40	+85	°C

12. Static characteristics

Table 14. Static characteristics

V_{DD} = 2.3 V to 5.5 V; V_{SS} = 0 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Supply						
V_{DD}	supply voltage		2.3	-	5.5	V
I_{DD}	supply current	operating mode; no load; f_{SCL} = 1 MHz; V_{DD} = 2.3 V to 5.5 V	-	6	10	mA
I_{stb}	standby current	no load; f_{SCL} = 0 Hz; V_I = V_{DD} or V_{SS}; V_{DD} = 2.3 V to 5.5 V	-	2.2	15.5	µA
V_{POR}	power-on reset voltage	no load; V_I = V_{DD} or V_{SS} [1]	-	1.70	2.0	V
Input SCL; input/output SDA						
V_{IL}	LOW-level input voltage		−0.5	-	+0.3V_{DD}	V
V_{IH}	HIGH-level input voltage		0.7V_{DD}	-	5.5	V
I_{OL}	LOW-level output current	V_{OL} = 0.4 V; V_{DD} = 2.3 V	20	28	-	mA
		V_{OL} = 0.4 V; V_{DD} = 5.0 V	30	40	-	mA
I_L	leakage current	V_I = V_{DD} or V_{SS}	−1	-	+1	µA
C_i	input capacitance	V_I = V_{SS}	-	6	10	pF
LED driver outputs						
I_{OL}	LOW-level output current	V_{OL} = 0.5 V; V_{DD} = 2.3 V to 4.5 V [2]	12	25	-	mA
$I_{OL(tot)}$	total LOW-level output current	V_{OL} = 0.5 V; V_{DD} = 4.5 V [2]	-	-	400	mA
I_{OH}	HIGH-level output current	open-drain; V_{OH} = V_{DD}	−10	-	+10	µA
V_{OH}	HIGH-level output voltage	I_{OH} = −10 mA; V_{DD} = 2.3 V	1.6	-	-	V
		I_{OH} = −10 mA; V_{DD} = 3.0 V	2.3	-	-	V
		I_{OH} = −10 mA; V_{DD} = 4.5 V	4.0	-	-	V
I_{OZ}	OFF-state output current	3-state; V_{OH} = V_{DD} or V_{SS}	−10	-	+10	µA
C_o	output capacitance		-	5	8	pF

Table 14. Static characteristics *...continued*

V_{DD} = 2.3 V to 5.5 V; V_{SS} = 0 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Address inputs; $\overline{OE}$ input; EXTCLK						
V_{IL}	LOW-level input voltage		−0.5	-	+0.3V_{DD}	V
V_{IH}	HIGH-level input voltage		0.7V_{DD}	-	5.5	V
I_{LI}	input leakage current		−1	-	+1	μA
C_i	input capacitance		-	3	5	pF

[1] V_{DD} must be lowered to 0.2 V in order to reset part.

[2] Each bit must be limited to a maximum of 25 mA and the total package limited to 400 mA due to internal busing limits.

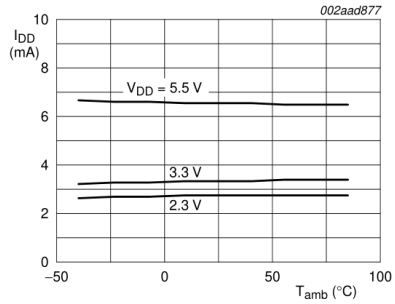


Fig 28. I_{DD} typical values with OSC on and f_{SCL} = 1 MHz versus temperature

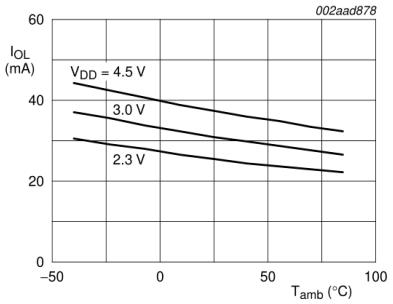


Fig 29. I_{OL} typical drive (LEDn outputs) versus temperature

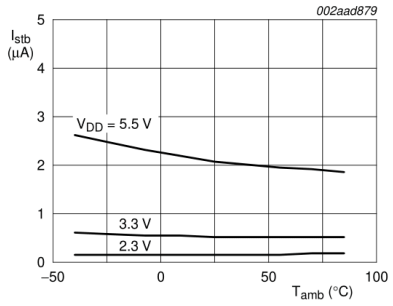


Fig 30. Standby supply current versus temperature

13. Dynamic characteristics

Table 15. Dynamic characteristics

Symbol	Parameter	Conditions	Standard-mode I²C-bus		Fast-mode I²C-bus		Fast-mode Plus I²C-bus		Unit
			Min	Max	Min	Max	Min	Max	
f_{SCL}	SCL clock frequency	[1]	0	100	0	400	0	1000	kHz
f_{EXTCLK}	frequency on pin EXTCLK		DC	50	DC	50	DC	50	MHz
t_{BUF}	bus free time between a STOP and START condition		4.7	-	1.3	-	0.5	-	µs
$t_{HD;STA}$	hold time (repeated) START condition		4.0	-	0.6	-	0.26	-	µs
$t_{SU;STA}$	set-up time for a repeated START condition		4.7	-	0.6	-	0.26	-	µs
$t_{SU;STO}$	set-up time for STOP condition		4.0	-	0.6	-	0.26	-	µs
$t_{HD;DAT}$	data hold time		0	-	0	-	0	-	ns
$t_{VD;ACK}$	data valid acknowledge time	[2]	0.3	3.45	0.1	0.9	0.05	0.45	µs
$t_{VD;DAT}$	data valid time	[3]	0.3	3.45	0.1	0.9	0.05	0.45	µs
$t_{SU;DAT}$	data set-up time		250	-	100	-	50	-	ns
t_{LOW}	LOW period of the SCL clock		4.7	-	1.3	-	0.5	-	µs
t_{HIGH}	HIGH period of the SCL clock		4.0	-	0.6	-	0.26	-	µs
t_f	fall time of both SDA and SCL signals	[4][5]	-	300	20 + 0.1C_b[6]	300	-	120	ns
t_r	rise time of both SDA and SCL signals		-	1000	20 + 0.1C_b[6]	300	-	120	ns
t_{SP}	pulse width of spikes that must be suppressed by the input filter	[7]	-	50	-	50	-	50	ns
t_{PLZ}	LOW to OFF-state propagation delay	$\overline{OE}$ to LEDn; OUTNE[1:0] = 10 or 11 in MODE2 register	-	40	-	40	-	40	ns
t_{PZL}	OFF-state to LOW propagation delay	$\overline{OE}$ to LEDn; OUTNE[1:0] = 10 or 11 in MODE2 register	-	60	-	60	-	60	ns
t_{PHZ}	HIGH to OFF-state propagation delay	$\overline{OE}$ to LEDn; OUTNE[1:0] = 10 or 11 in MODE2 register	-	60	-	60	-	60	ns

Table 15. Dynamic characteristics *…continued*

Symbol	Parameter	Conditions	Standard-mode I²C-bus		Fast-mode I²C-bus		Fast-mode Plus I²C-bus		Unit
			Min	Max	Min	Max	Min	Max	
t_{PZH}	OFF-state to HIGH propagation delay	$\overline{OE}$ to LEDn; OUTNE[1:0] = 10 or 11 in MODE2 register	-	40	-	40	-	40	ns
t_{PLH}	LOW to HIGH propagation delay	$\overline{OE}$ to LEDn; OUTNE[1:0] = 01 in MODE2 register	-	40	-	40	-	40	ns
t_{PHL}	HIGH to LOW propagation delay	$\overline{OE}$ to LEDn; OUTNE[1:0] = 00 in MODE2 register	-	60	-	60	-	60	ns

[1] Minimum SCL clock frequency is limited by the bus time-out feature, which resets the serial bus interface if either SDA or SCL is held LOW for a minimum of 25 ms. Disable bus time-out feature for DC operation.

[2] $t_{VD;ACK}$ = time for Acknowledgement signal from SCL LOW to SDA (out) LOW.

[3] $t_{VD;DAT}$ = minimum time for SDA data out to be valid following SCL LOW.

[4] A master device must internally provide a hold time of at least 300 ns for the SDA signal (refer to the V_{IL} of the SCL signal) in order to bridge the undefined region of SCL's falling edge.

[5] The maximum t_f for the SDA and SCL bus lines is specified at 300 ns. The maximum fall time (t_f) for the SDA output stage is specified at 250 ns. This allows series protection resistors to be connected between the SDA and the SCL pins and the SDA/SCL bus lines without exceeding the maximum specified t_f.

[6] C_b = total capacitance of one bus line in pF.

[7] Input filters on the SDA and SCL inputs suppress noise spikes less than 50 ns.

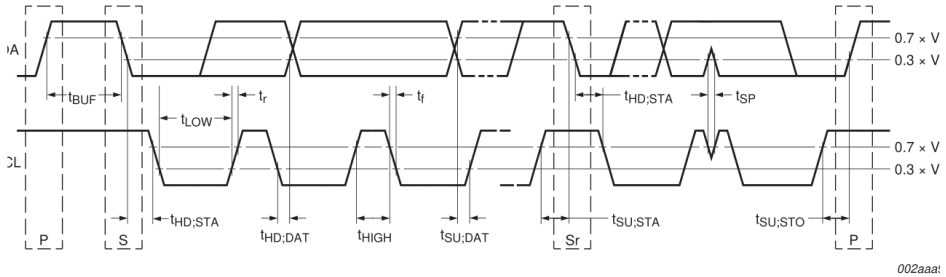
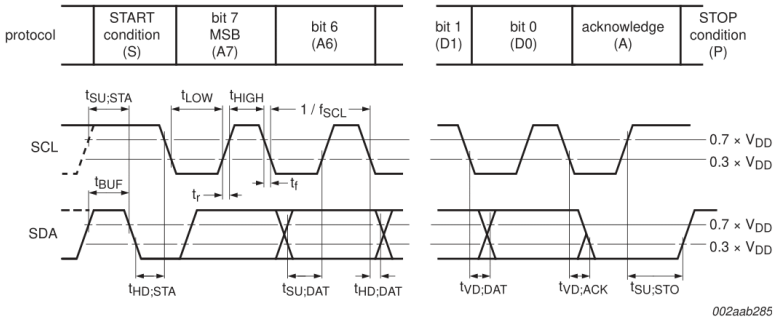


Fig 31. Definition of timing



Rise and fall times refer to V_{IL} and V_{IH}.

Fig 32. I²C-bus timing diagram

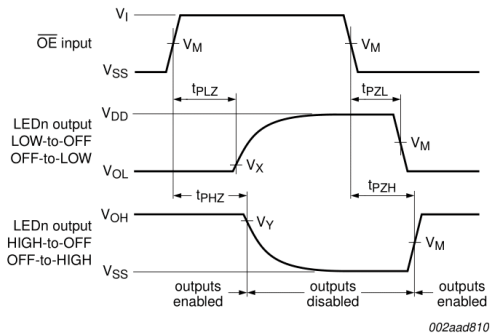
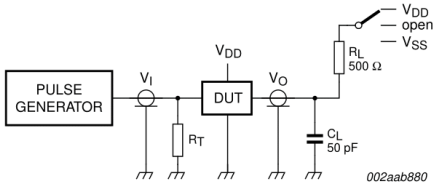


Fig 33. t_{PLZ}, t_{PZL} and t_{PHZ}, t_{PZH} times

14. Test information

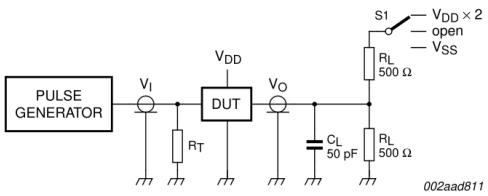


R_L = Load resistor for LEDn.

C_L = Load capacitance includes jig and probe capacitance.

R_T = Termination resistance should be equal to the output impedance Z_o of the pulse generators.

Fig 34. Test circuitry for switching times



R_L = Load resistor for LEDn.

C_L = Load capacitance includes jig and probe capacitance.

R_T = Termination resistance should be equal to the output impedance Z_o of the pulse generators.

Test data are given in Table 16.

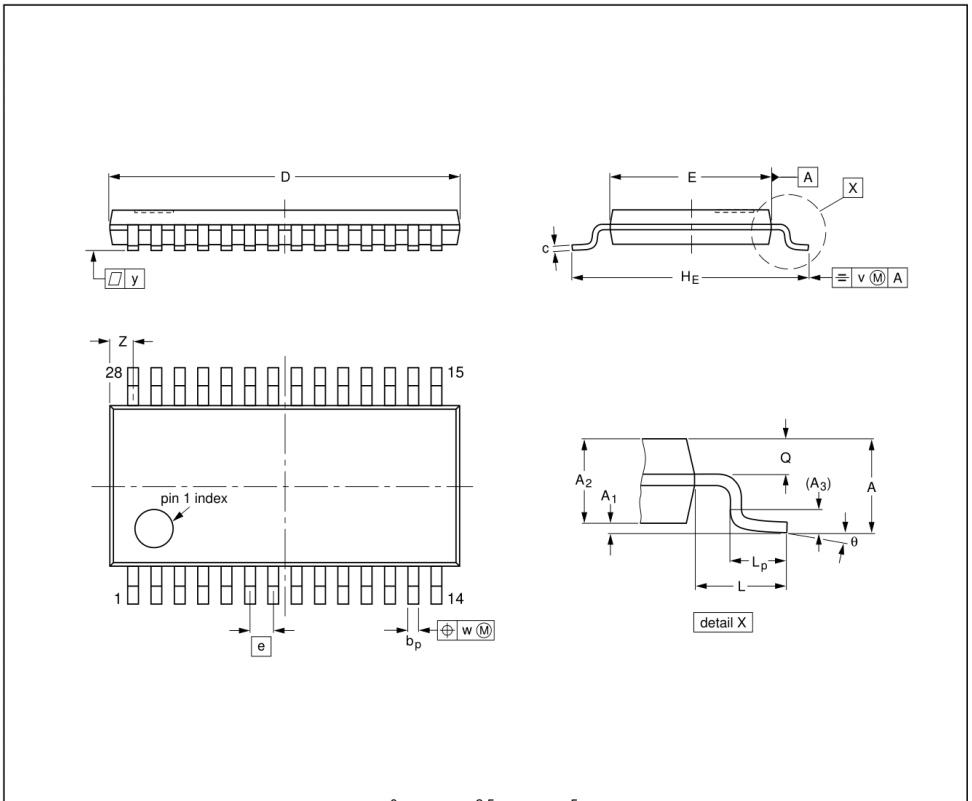
Fig 35. Test circuitry for switching times for enable/disable

Table 16. Test data for enable/disable switching times

Test	Load		Switch
	C_L	R_L	
t_{PD}	50 pF	500 Ω	open
t_{PLZ}, t_{PZL}	50 pF	500 Ω	$V_{DD} \times 2$
t_{PHZ}, t_{PZH}	50 pF	500 Ω	V_{SS}

15. Package outline

TSSOP28: plastic thin shrink small outline package; 28 leads; body width 4.4 mm **SOT361-1**



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	D (1)	E (2)	e	H_E	L	L_p	Q	v	w	y	Z (1)	θ
mm	1.1	0.15 0.05	0.95 0.80	0.25	0.30 0.19	0.2 0.1	9.8 9.6	4.5 4.3	0.65	6.6 6.2	1	0.75 0.50	0.4 0.3	0.2	0.13	0.1	0.8 0.5	8° 0°

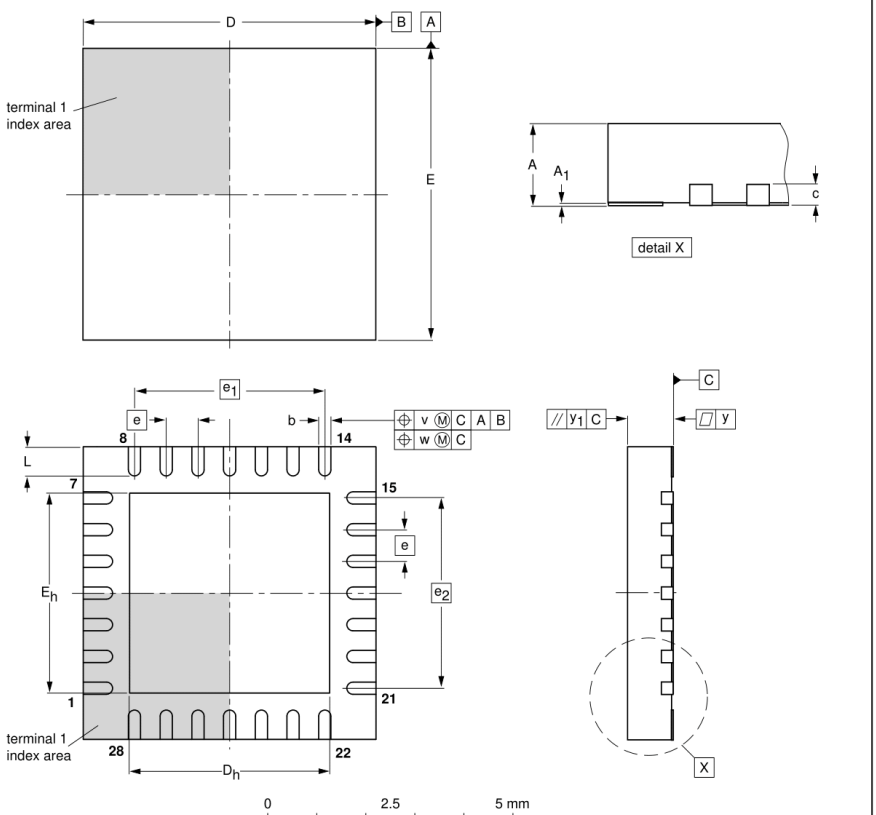
Notes

1. Plastic or metal protrusions of 0.15 mm maximum per side are not included.
2. Plastic interlead protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT361-1		MO-153				~~99-12-27~~ 03-02-19

Fig 36. Package outline SOT361-1 (TSSOP28)

HVQFN28: plastic thermal enhanced very thin quad flat package; no leads; 28 terminals; body 6 x 6 x 0.85 mm **SOT788-1**



DIMENSIONS (mm are the original dimensions)

UNIT	A(1) max.	A_1	b	c	D(1)	D_h	E(1)	E_h	e	e_1	e_2	L	v	w	y	y_1
mm	1	0.05 0.00	0.35 0.25	0.2	6.1 5.9	4.25 3.95	6.1 5.9	4.25 3.95	0.65	3.9	3.9	0.75 0.50	0.1	0.05	0.05	0.1

Note

1. Plastic or metal protrusions of 0.075 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT788-1	- - -	MO-220	- - -			02-10-22

Fig 37. Package outline SOT788-1 (HVQFN28)

16. Handling information

All input and output pins are protected against ElectroStatic Discharge (ESD) under normal handling. When handling ensure that the appropriate precautions are taken as described in *JESD625-A* or equivalent standards.

17. Soldering of SMD packages

This text provides a very brief insight into a complex technology. A more in-depth account of soldering ICs can be found in Application Note *AN10365 "Surface mount reflow soldering description"*.

17.1 Introduction to soldering

Soldering is one of the most common methods through which packages are attached to Printed Circuit Boards (PCBs), to form electrical circuits. The soldered joint provides both the mechanical and the electrical connection. There is no single soldering method that is ideal for all IC packages. Wave soldering is often preferred when through-hole and Surface Mount Devices (SMDs) are mixed on one printed wiring board; however, it is not suitable for fine pitch SMDs. Reflow soldering is ideal for the small pitches and high densities that come with increased miniaturization.

17.2 Wave and reflow soldering

Wave soldering is a joining technology in which the joints are made by solder coming from a standing wave of liquid solder. The wave soldering process is suitable for the following:

- Through-hole components
- Leaded or leadless SMDs, which are glued to the surface of the printed circuit board

Not all SMDs can be wave soldered. Packages with solder balls, and some leadless packages which have solder lands underneath the body, cannot be wave soldered. Also, leaded SMDs with leads having a pitch smaller than ~0.6 mm cannot be wave soldered, due to an increased probability of bridging.

The reflow soldering process involves applying solder paste to a board, followed by component placement and exposure to a temperature profile. Leaded packages, packages with solder balls, and leadless packages are all reflow solderable.

Key characteristics in both wave and reflow soldering are:

- Board specifications, including the board finish, solder masks and vias
- Package footprints, including solder thieves and orientation
- The moisture sensitivity level of the packages
- Package placement
- Inspection and repair
- Lead-free soldering versus SnPb soldering

17.3 Wave soldering

Key characteristics in wave soldering are:

- Process issues, such as application of adhesive and flux, clinching of leads, board transport, the solder wave parameters, and the time during which components are exposed to the wave
- Solder bath specifications, including temperature and impurities

17.4 Reflow soldering

Key characteristics in reflow soldering are:

- Lead-free versus SnPb soldering; note that a lead-free reflow process usually leads to higher minimum peak temperatures (see Figure 38) than a SnPb process, thus reducing the process window
- Solder paste printing issues including smearing, release, and adjusting the process window for a mix of large and small components on one board
- Reflow temperature profile; this profile includes preheat, reflow (in which the board is heated to the peak temperature) and cooling down. It is imperative that the peak temperature is high enough for the solder to make reliable solder joints (a solder paste characteristic). In addition, the peak temperature must be low enough that the packages and/or boards are not damaged. The peak temperature of the package depends on package thickness and volume and is classified in accordance with Table 17 and 18

Table 17. SnPb eutectic process (from J-STD-020D)

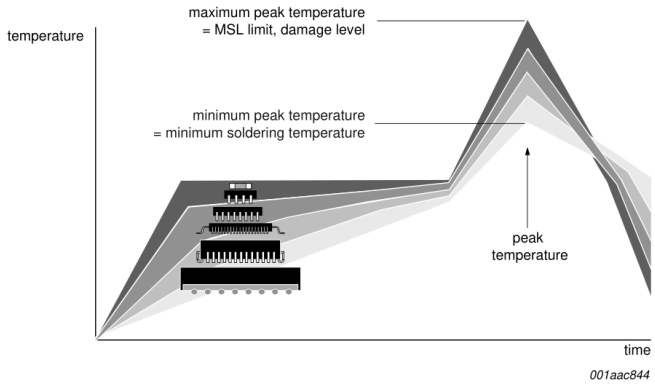
Package thickness (mm)	Package reflow temperature (°C)	
	Volume (mm³)	
	< 350	≥ 350
< 2.5	235	220
≥ 2.5	220	220

Table 18. Lead-free process (from J-STD-020D)

Package thickness (mm)	Package reflow temperature (°C)		
	Volume (mm³)		
	< 350	350 to 2000	> 2000
< 1.6	260	260	260
1.6 to 2.5	260	250	245
> 2.5	250	245	245

Moisture sensitivity precautions, as indicated on the packing, must be respected at all times.

Studies have shown that small packages reach higher temperatures during reflow soldering, see Figure 38.



MSL: Moisture Sensitivity Level

Fig 38. Temperature profiles for large and small components

For further information on temperature profiles, refer to Application Note *AN10365 "Surface mount reflow soldering description"*.

18. Abbreviations

Table 19. Abbreviations

Acronym	Description
CDM	Charged-Device Model
DUT	Device Under Test
EMI	ElectroMagnetic Interference
ESD	ElectroStatic Discharge
HBM	Human Body Model
I²C-bus	Inter-Integrated Circuit bus
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LSB	Least Significant Bit
MM	Machine Model
MSB	Most Significant Bit
NMOS	Negative-channel Metal-Oxide Semiconductor
PCB	Printed-Circuit Board
PMOS	Positive-channel Metal-Oxide Semiconductor
POR	Power-On Reset
PWM	Pulse Width Modulation; Pulse Width Modulator
RGB	Red/Green/Blue
RGBA	Red/Green/Blue/Amber
SMBus	System Management Bus

19. Revision history

Table 20. Revision history

Document ID	Release date	Data sheet status	Change notice	Supersedes
PCA9685 v.4	20150416	Product data sheet	-	PCA9685 v.3
Modifications:	• Changed programmable frequency to “24 Hz to 1526 Hz” throughout • Minor edits to text and figures to provide clarity regarding cycle count throughout			
PCA9685 v.3	20100902	Product data sheet	-	PCA9685 v.2
PCA9685 v.2	20090716	Product data sheet	-	PCA9685 v.1
PCA9685 v.1	20080724	Product data sheet	-	-

20. Legal information

20.1 Data sheet status

Document status[1][2]	Product status[3]	Definition
Objective [short] data sheet	Development	This document contains data from the objective specification for product development.
Preliminary [short] data sheet	Qualification	This document contains data from the preliminary specification.
Product [short] data sheet	Production	This document contains the product specification.

[1] Please consult the most recently issued document before initiating or completing a design.

[2] The term 'short data sheet' is explained in section "Definitions".

[3] The product status of device(s) described in this document may have changed since this document was published and may differ in case of multiple devices. The latest product status information is available on the Internet at URL http://www.nxp.com.

20.2 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

Short data sheet — A short data sheet is an extract from a full data sheet with the same product type number(s) and title. A short data sheet is intended for quick reference only and should not be relied upon to contain detailed and full information. For detailed and full information see the relevant full data sheet, which is available on request via the local NXP Semiconductors sales office. In case of any inconsistency or conflict with the short data sheet, the full data sheet shall prevail.

Product specification — The information and data provided in a Product data sheet shall define the specification of the product as agreed between NXP Semiconductors and its customer, unless NXP Semiconductors and customer have explicitly agreed otherwise in writing. In no event however, shall an agreement be valid in which the NXP Semiconductors product is deemed to offer functions and qualities beyond those described in the Product data sheet.

20.3 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

20.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

21. Contact information

For more information, please visit: **http://www.nxp.com**

For sales office addresses, please send an email to: **salesaddresses@nxp.com**

22. Contents

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 16 April 2015
Document identifier: PCA9685

Appendix K – SRD-05VDC Relay Datasheet

SONGLE RELAY

松乐继电器 ® SONGLE RELAY	RELAY ISO9002	SRD

1. MAIN FEATURES

- Switching capacity available by 10A in spite of small size design for highdensity P.C. board mounting technique.
- UL,CUL,TUV recognized.
- Selection of plastic material for high temperature and better chemical solution performance.
- Sealed types available.
- Simple relay magnetic circuit to meet low cost of mass production.

2. APPLICATIONS

- Domestic appliance, office machine, audio, equipment, automobile, etc.
  (Remote control TV receiver, monitor display, audio equipment high rushing current use application.)

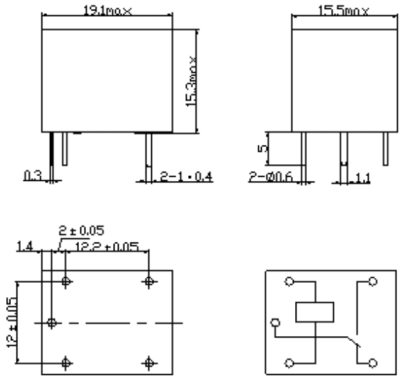
3. ORDERING INFORMATION

SRD	XX VDC	S	L	C
Model of relay	Nominal coil voltage	Structure	Coil sensitivity	Contact form
SRD	03、05、06、09、12、24、48VDC	S:Sealed type	L:0.36W	A:1 form A
				B:1 form B
		F:Flux free type	D:0.45W	C:1 form C

4. RATING

CCC	FILE NUMBER:CH0052885-2000	7A/240VDC
CCC	FILE NUMBER:CH0036746-99	10A/250VDC
UL /CUL	FILE NUMBER: E167996	10A/125VAC 28VDC
TUV	FILE NUMBER: R9933789	10A/240VAC 28VDC

5. DIMENSION(unit:mm) DRILLING(unit:mm) WIRING DIAGRAM



6. COIL DATA CHART (AT20°C)

Coil Sensitivity	Coil Voltage Code	Nominal Voltage (VDC)	Nominal Current (mA)	Coil Resistance (Ω) ±10%	Power Consumption (W)	Pull-In Voltage (VDC)	Drop-Out Voltage (VDC)	Max-Allowable Voltage (VDC)
SRD (High Sensitivity)	03	03	120	25	abt. 0.36W	75%Max.	10% Min.	120%
	05	05	71.4	70				
	06	06	60	100				
	09	09	40	225				
	12	12	30	400				
	24	24	15	1600				
	48	48	7.5	6400				
SRD (Standard)	03	03	150	20	abt. 0.45W	75% Max.	10% Min.	110%
	05	05	89.3	55				
	06	06	75	80				
	09	09	50	180				
	12	12	37.5	320				
	24	24	18.7	1280				
	48	48	10	4500	abt. 0.51W			

7. CONTACT RATING

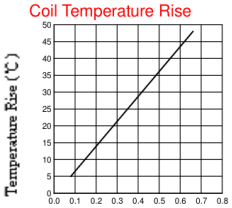
Item \ Type	SRD	
	FORM C	FORM A
Contact Capacity Resistive Load (cosΦ=1)	7A 28VDC 10A 125VAC 7A 240VAC	10A 28VDC 10A 240VAC
Inductive Load (cosΦ=0.4 L/R=7msec)	3A 120VAC 3A 28VDC	5A 120VAC 5A 28VDC
Max. Allowable Voltage	250VAC/110VDC	250VAC/110VDC
Max. Allowable Power Force	800VAC/240W	1200VA/300W
Contact Material	AgCdO	AgCdO

8. PERFORMANCE (at initial value)

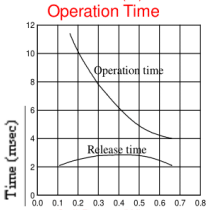
Item \ Type	SRD
Contact Resistance	100mΩ Max.
Operation Time	10msec Max.
Release Time	5msec Max.
Dielectric Strength Between coil & contact Between contacts	 1500VAC 50/60HZ (1 minute) 1000VAC 50/60HZ (1 minute)
Insulation Resistance	100 MΩ Min. (500VDC)
Max. ON/OFF Switching Mechanically Electrically	 300 operation/min 30 operation/min
Ambient Temperature	-25°C to +70°C
Operating Humidity	45 to 85% RH
Vibration Endurance Error Operation	 10 to 55Hz Double Amplitude 1.5mm 10 to 55Hz Double Amplitude 1.5mm
Shock Endurance Error Operation	 100G Min. 10G Min.
Life Expectancy Mechanically Electrically	 10^7 operations. Min. (no load) 10^5 operations. Min. (at rated coil voltage)
Weight	abt. 10grs.

9.REFERENCE DATA

Coil Temperature Rise

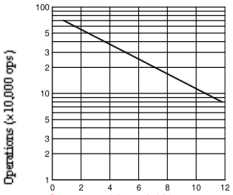


Operation Time

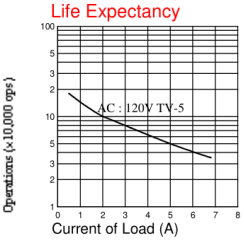


Life Expectancy

AC120V/DC24V cosΦ=1



Life Expectancy



Appendix L – Talentcell 12v Battery Datasheet

封面

TalentCell

User manual of 12V Lithium ion power bank

Any questions, please feel free to contact us service@talentcell.com

Product specification

Model	YB1203000	YB1206000	YB1208300-USB
Photos			
Size(mm)	25*65*105	28*85*145	39*80*138
Weight(g)	190	360	500
12V Capacity(mAh)	3000	6000	8300
Input	12.6V 3A max.	12.6V 3A max.	12.6V 6A max.
Output	12(9-12.6)V/3A max.	12(9-12.6)V/3A max.	12V(9-12.6)/6A max. 9V/1A 5V/2A
Spec. of charger	12.6V 0.5A	12.6V 1A	12.6V 1.5A
Charge Time	~6hrs	~6hrs	~6hrs
Operate Temperature	-20~60℃	-20~60℃	-20~60℃
Fuel Gauge	Yes	Yes	Yes

Model	YB1203000-USB	YB1206000-USB	YB12011000-USB
Photos			
Size(mm)	25*65*105	28*85*145	39*80*138
Weight(g)	190	360	505
12V Capacity(mAh)	3000	6000	11000
Input	12.6V 3A max.	12.6V 3A max.	12.6V 6A max.
Output	12(9-12.6)V/3A max. 5V/2A	12(9-12.6)V/3A max. 5V/2A	12(9-12.6)V/6A max. 9V/1A 5V/2A
Spec. of charger	12.6V 0.5A	12.6V 1A	12.6V 1.5A
Charge Time	~6hrs	~6hrs	~7hrs
Operate Temperature	-20~60℃	-20~60℃	-20~60℃
Fuel Gauge	Yes	Yes	Yes

YB1203000

YB1203000-USB

YB1206000

YB1206000-USB

YB1208300-USB

YB12011000-USB

Operation guidance

1.Power ON and OFF
ON: Push switch to Position "I" to make power on
OFF: Push switch to Position "O" to make power off

2.Capacity display(Fuel Gauge): There are 5 power indicators, YB1208300-USB and YB12011000-USB have 4 power indicators. Push the switch to Position "I" to make battery power on. The power and capacity indicator lights will be lighted.

How to charge the power bank

1. Connect the AC-DC charger to the power socket(100-240V available)
2. Push the switch of the power bank to position"I" to make the charging circuit breakover.
3. Connect the power bank to the AC-DC charger .
4. The LED indicator of the AC-DC charger showing RED means the charging process is working.
The LED indicator of the AC-DC charger showing GREEN means the charging process has completed or the charging circuit is open.
5. The LED indicator of the AC-DC charger will change from RED to GREEN while the power bank is charged with 95% of the full.
6. After the unit is fully charged, please remove the AC wall charger in time.
7. Full charge time (estimated)=Capacity of power bank / output current of charger

Compatible with

Compatible with LED strip light products, CCTV Camera, IP Camera, Telescopes, LED Panel, Amplifier, Modem, Car DVR, Speaker, mobile phone, breast pump portable and most 12V devices, etc.

Accessories

1x Power bank
1x DC 5.5x2.1mm 1 female to 2 male power splitter cable
1x 12.6V AC-DC charger

Note: only the YB12011000-USB and YB1208300-USB packages include the DC 5521 male to 5525 male cable.

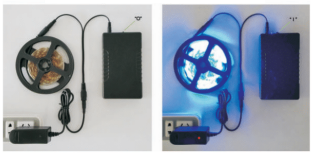
DC 5.5x2.1mm 1 Female to 2 Male Power Splitter Cable

DC 5.5x2.1mm 1 Male to 5.5x2.5mm 1 Male Cable
Use for YB1208300-USB and YB12011000-USB

12.6V AC-DC Charger

How to use the 1 Female to 2 Male cable

You can output while charging at the same time.



Cautions

1.The battery should be taken to avoid exposure, percussion, water, and rain in the transport process, and make the battery positive and negative poles insulated.
2.Do not long-term placement in a low voltage state. In order to extend the service life, it is suggested charge every three months.
3.Do not put the battery into fire or water.
4.Do not modify or disassemble the battery.
5.Do not impact heavily and throw the battery.
6.Be careful not to let children touch the battery.
7.Do not reverse connect the input or the output.
8.Please use the qualified charger to charge the battery.
9.The battery should be stored in a cool dry indoor storage, avoid flooding, rain.
10.If there is any abnormal or malfunction at the time of charging, please contact your dealer.

Contact us

For any inquiries or comments concerning our product or any quality-related issue with our product, or you want to customize any other battery packs, please feel free to contact us, we will respond to you as soon as possible and do our best to help you.

Email us: **service@talentcell.com**
Website: **www.talentcell.com**

Producted by TalentCell

Customized products

Customize batteries for your business, please feel free to contact us.

Dear Friend

Thanks for choosing TalentCell.
We are specialized in manufacturing rechargeable batteries and premium quality consumer electronic products and accessories since year of 2001.
We strive to offer the 100% satisfaction for all our customers. if you have any questions. please don't hesitate to contact us. we'll be sure to solve issues as soon as possible.

Best Regards! Have a nice day!
TalentCell Technology

Appendix M – XL4015 DC-DC Step-down Buck converter Datasheet

5A 180KHz 36V Buck DC to DC Converter **XL4015**

Features

- Wide 8V to 36V Input Voltage Range
- Output Adjustable from 1.25V to 32V
- Maximum Duty Cycle 100%
- Minimum Drop Out 0.3V
- Fixed 180KHz Switching Frequency
- 5A Constant Output Current Capability
- Internal Optimize Power MOSFET
- High efficiency up to 96%
- Excellent line and load regulation
- Built in thermal shutdown function
- Built in current limit function
- Built in output short protection function
- Available in TO263-5L package

Applications

- LCD Monitor and LCD TV
- Portable instrument power supply
- Telecom / Networking Equipment

General Description

The XL4015 is a 180 KHz fixed frequency PWM buck (step-down) DC/DC converter, capable of driving a 5A load with high efficiency, low ripple and excellent line and load regulation. Requiring a minimum number of external components, the regulator is simple to use and include internal frequency compensation and a fixed-frequency oscillator.

The PWM control circuit is able to adjust the duty ratio linearly from 0 to 100%. An over current protection function is built inside. When short protection function happens, the operation frequency will be reduced from 180KHz to 48KHz. An internal compensation block is built in to minimize external component count.

Figure1. Package Type of XL4015

Pin Configurations

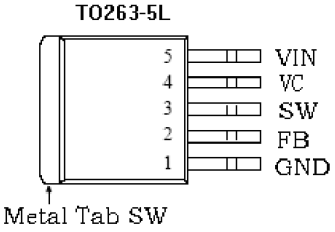


Figure2. Pin Configuration of XL4015 (Top View)

Table 1 Pin Description

Pin Number	Pin Name	Description
1	GND	Ground Pin. Care must be taken in layout. This pin should be placed outside of the Schottky Diode to output capacitor ground path to prevent switching current spikes from inducing voltage noise into XL4015.
2	FB	Feedback Pin (FB). Through an external resistor divider network, FB senses the output voltage and regulates it. The feedback threshold voltage is 1.25V.
3	SW	Power Switch Output Pin (SW). SW is the switch node that supplies power to the output.
4	VC	Internal Voltage Regulator Bypass Capacity. In typical system application, The VC pin connect a 1uf capacity to VIN.
5	VIN	Supply Voltage Input Pin. XL4015 operates from a 8V to 36V DC voltage. Bypass Vin to GND with a suitably large capacitor to eliminate noise on the input.

Function Block

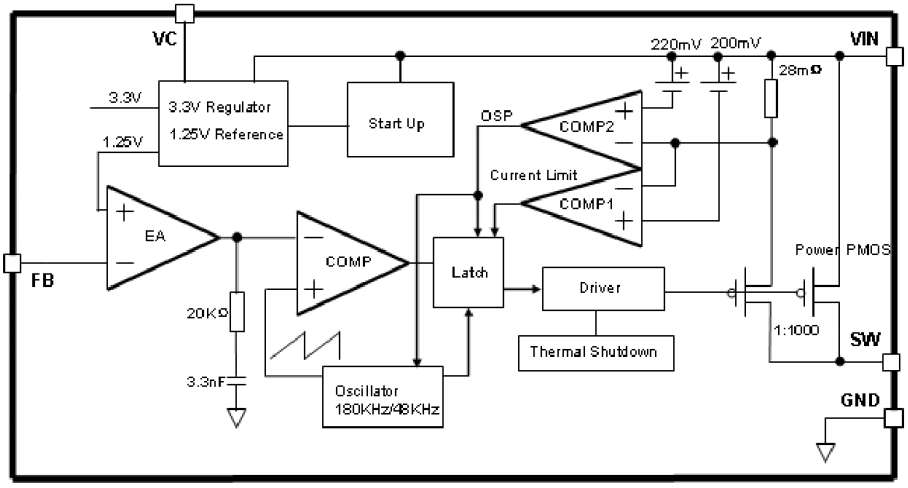


Figure3. Function Block Diagram of XL4015

Typical Application Circuit

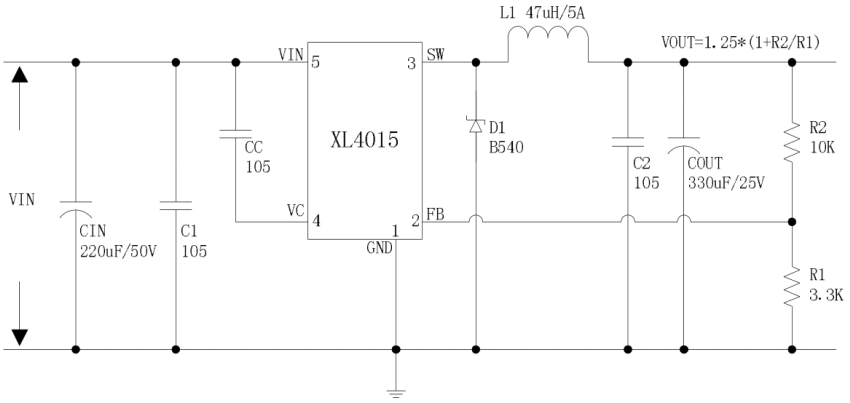


Figure4. XL4015 Typical Application Circuit (VIN=8V~36V, VOUT=5V/5A)

Ordering Information

Package	Temperature Range	Part Number	Marking ID	Packing Type
		Lead Free	Lead Free	
		XL4015E1	XL4015E1	800 Units on Tape & Reel

XLSEMI Pb-free products, as designated with "E1" suffix in the par number, are RoHS compliant.

Absolute Maximum Ratings（Note1）

Parameter	Symbol	Value	Unit
Input Voltage	Vin	-0.3 to 40	V
Feedback Pin Voltage	V_{FB}	-0.3 to Vin	V
Output Switch Pin Voltage	V_{Output}	-0.3 to Vin	V
Power Dissipation	P_D	Internally limited	mW
Thermal Resistance (TO263-5L) (Junction to Ambient, No Heatsink, Free Air)	R_{JA}	30	ºC/W
Operating Junction Temperature	T_J	-40 to 125	ºC
Storage Temperature	T_{STG}	-65 to 150	ºC
Lead Temperature (Soldering, 10 sec)	T_{LEAD}	260	ºC
ESD (HBM)		>2000	V

Note1: Stresses greater than those listed under Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.

XL4015 Electrical Characteristics

T_a = 25℃;unless otherwise specified.

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Unit
System parameters test circuit figure4						
VFB	Feedback Voltage	Vin = 8V to 40V, Vout=5V Iload=0.5A to 5A	1.225	1.25	1.275	V
Efficiency	ŋ	Vin=12V ,Vout=5V Iout=5A	-	87	-	%
Efficiency	ŋ	Vin=24V ,Vout=12V Iout=5A	-	93	-	%

Electrical Characteristics (DC Parameters)

Vin = 12V, GND=0V, Vin & GND parallel connect a 220uf/50V capacitor; Iout=500mA, T_a = 25℃; the others floating unless otherwise specified.

Parameters	Symbol	Test Condition	Min.	Typ.	Max.	Unit
Input operation voltage	Vin		8		36	V
Quiescent Supply Current	I_q	V_{FB} =Vin		2.1	5	mA
Oscillator Frequency	Fosc		144	180	216	KHz
Output Short Frequency	Fosp			48		KHz
Switch Current Limit	I_L	V_{FB} =0		7		A
Max. Duty Cycle	D_{MAX}	V_{FB}=0V		100		%
Output Power PMOS	Rdson	V_{FB}=0V, Vin=12V, I_{SW}=5A		60	80	mohm

Typical System Application (VOUT=5V/5A)

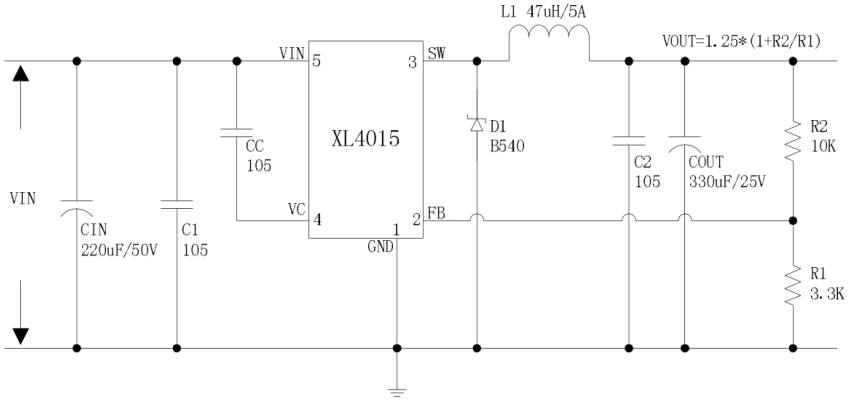


Figure5. XL4015 System Parameters Test Circuit (VIN=8V~36V, VOUT=5V/5A)

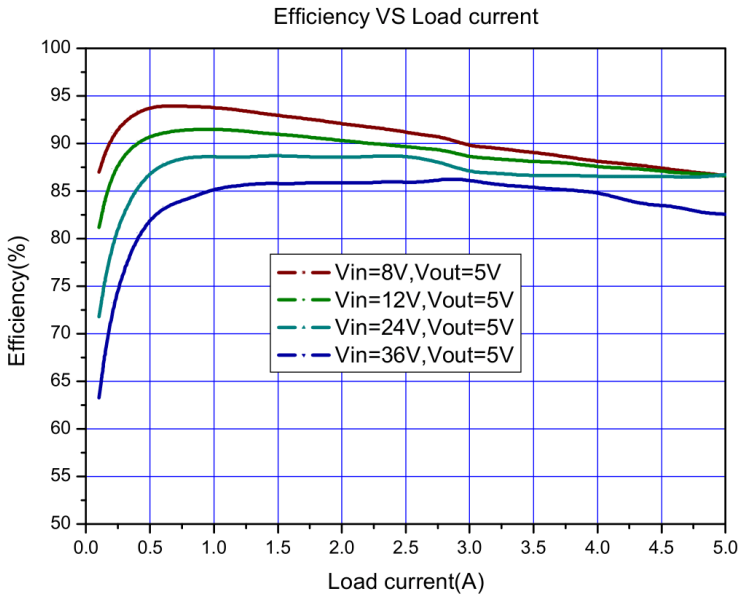


Figure6. XL4015 System Efficiency Curve

Typical System Application (VOUT=12V/5A)

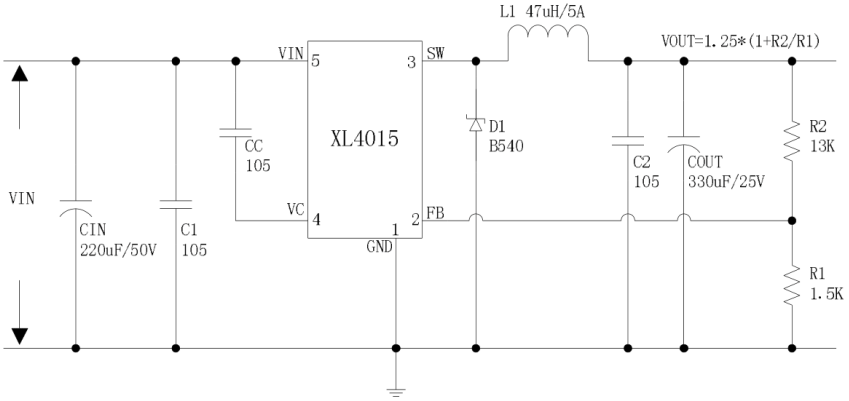


Figure7. XL4015 System Parameters Test Circuit (VIN=15V~36V, VOUT=12V/5A)

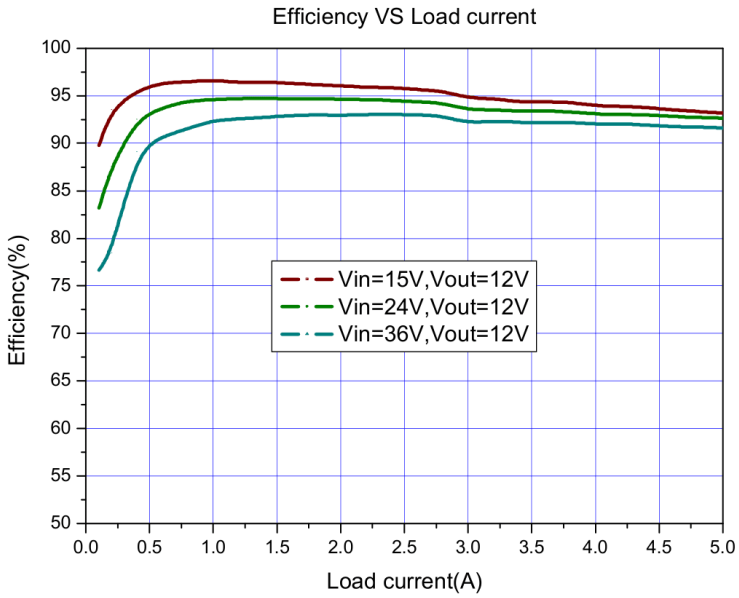


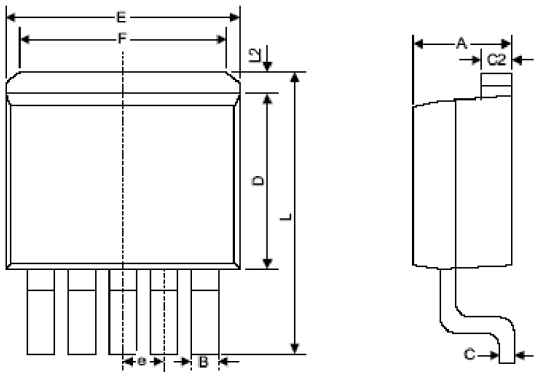
Figure8. XL4015 System Efficiency Curve

Schottky Diode Selection Table

Current	Surface Mount	Through Hole	VR (The same as system maximum input voltage)				
			20V	30V	40V	50V	60V
1A		√	1N5817	1N5818	1N5819		
3A		√	1N5820	1N5821	1N5822		
		√	MBR320	MBR330	MBR340	MBR350	MBR360
	√		SK32	SK33	SK34	SK35	SK36
	√			30WQ03	30WQ04	30WQ05	
		√		31DQ03	31DQ04	31DQ05	
		√	SR302	SR303	SR304	SR305	SR306
5A		√	1N5823	1N5824	1N5825		
		√	SR502	SR503	SR504	SR505	SR506
		√	SB520	SB530	SB540	SB550	SB560
	√		SK52	SK53	SK54	SK55	SK56
	√			50WQ03	50WQ04	50WQ05	

Package Information

TO263-5L



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min.	Max.	Min.	Max.
A	4.06	4.83	0.160	0.190
B	0.76	1.02	0.030	0.040
C	0.36	0.64	0.014	0.025
C2	1.14	1.40	0.045	0.055
D	8.64	9.65	0.340	0.380
E	9.78	10.54	0.385	0.415
e	1.57	1.85	0.062	0.073
F	6.60	7.11	0.260	0.280
L	15.11	15.37	0.595	0.605
L2	-	1.40	-	0.055